

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **2000134229 A**(43) Date of publication of application: **12 . 05 . 00**

(51) Int. Cl

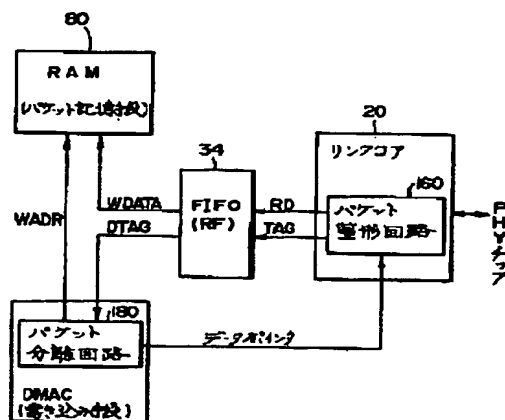
H04L 12/40(21) Application number: **10321488**(22) Date of filing: **27 . 10 . 98**(71) Applicant: **SEIKO EPSON CORP**(72) Inventor: **KANBARA YOSHIYUKI
ISHIDA TAKUYA
WADA FUMITOSHI**(54) **DATA TRANSFER CONTROLLER AND
ELECTRONIC EQUIPMENT**

COPYRIGHT: (C)2000,JPO

(57) Abstract:

PROBLEM TO BE SOLVED: To provide a data transfer controller and electronic equipment capable of reducing the overheads of processings and transferring data at high speed with the hardware of a small scale.

SOLUTION: Relating to this data transfer controller of IEEE1394, a packet shaping circuit 160 shapes packets transferred from respective nodes so as to be used by an upper layer and a packet separation circuit 180 writes the headers of the shaped packets in the header area of a RAM and writes the data in a data area. Then, at the time of packet shaping, a data pointer delivered from the packet separation circuit 180 is added to the header of the packet. The packet is separated by using TAG. Broadcast information, error status information and information for indicating whether or not it is the packet received in a self ID period are added to the trailer of the packet at the time of packet shaping. The information of ACK or the like time sequentially added to the rear side of the packet in packet shaping is written to the head side of the header of the packet in the RAM 80.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-134229

(P2000-134229A)

(43) 公開日 平成12年5月12日 (2000.5.12)

(51) Int.Cl.⁷

識別記号

F I

テーマコード(参考)

H 0 4 L 12/40

H 0 4 L 11/00

3 2 0

5 K 0 3 2

審査請求 未請求 請求項の数13 F D (全 27 頁)

(21) 出願番号 特願平10-321488
(22) 出願日 平成10年10月27日 (1998. 10. 27)

(71) 出願人 000002369
セイコーエプソン株式会社
東京都新宿区西新宿2丁目4番1号
(72) 発明者 神原 義幸
長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内
(72) 発明者 石田 卓也
長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内
(74) 代理人 100090479
弁理士 井上 一 (外2名)

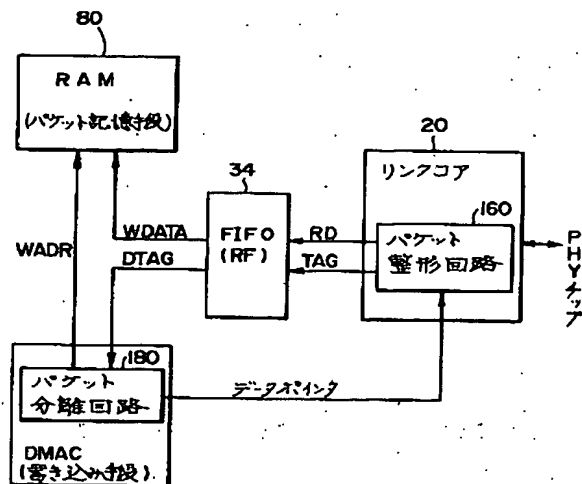
最終頁に続く

(54) 【発明の名称】 データ転送制御装置及び電子機器

(57) 【要約】

【課題】 処理のオーバーヘッドを軽減し、小規模なハードウェアで高速なデータ転送を実現できるデータ転送制御装置及び電子機器を提供すること。

【解決手段】 IEEE1394のデータ転送制御装置において、パケット整形回路160が、各ノードから転送されてきたパケットを上層が使用できるように整形し、パケット分離回路180が、整形されたパケットのヘッダをRAMのヘッダ領域に書き込み、データをデータ領域に書き込む。そして、パケット整形の際に、パケット分離回路から渡されたデータポイントをパケットのヘッダに付加する。TAGを用いてパケットを分離する。ブロードキャスト情報、エラーステータス情報、セルフID期間に受信したパケットか否かを示す情報を、パケット整形の際にパケットのトレーラーに付加する。パケット整形において時系列でパケットの後ろ側に付加されたACKなどの情報を、RAMにおいて、パケットのヘッダの先頭側に書き込む。



【特許請求の範囲】

【請求項 1】 バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、各ノードから転送されてきたパケットを上層が使用できるように整形するパケット整形手段と、整形されたパケットの制御情報をランダムアクセス可能な記憶手段の制御情報領域に書き込み、整形されたパケットのデータを前記記憶手段のデータ領域に書き込むパケット分離手段とを含み、前記パケット分離手段が、前記データ領域に書き込まれるデータのアドレスを指すデータポインタを前記パケット整形手段に渡し、前記パケット整形手段が、前記パケット分離手段から渡された前記データポインタをパケットの制御情報に付加することを特徴とするデータ転送制御装置。

【請求項 2】 請求項 1 において、前記パケット整形手段が、パケットの少なくとも制御情報とデータを区別するためのタグ情報を生成すると共に、生成された該タグ情報をパケットに関連づけ、前記パケット分離手段が、パケットに関連づけられた前記タグ情報に基づいて、パケットの制御情報を前記制御情報領域に書き込み、パケットのデータを前記データ領域に書き込むことを特徴とするデータ転送制御装置。

【請求項 3】 請求項 2 において、前記パケット分離手段が、パケットの制御情報が書き込まれていると、パケットに関連づけられた前記タグ情報に基づき判断した場合には、前記制御情報領域に書き込まれる制御情報のアドレスを指す制御情報ポインタを更新し、パケットのデータが書き込まれていると、前記タグ情報に基づき判断した場合には、前記データ領域に書き込まれるデータのアドレスを指すデータポインタを更新することを特徴とするデータ転送制御装置。

【請求項 4】 バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、各ノードから転送されてきたパケットを上層が使用できるように整形するパケット整形手段と、整形されたパケットを記憶手段に書き込む書き込み手段とを含み、前記パケット整形手段が、バスに接続される全てのノードに対して送信されたパケットか否かを示すブロードキャスト情報を、パケットの制御情報に付加することを特徴とするデータ転送制御装置。

【請求項 5】 バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、各ノードから転送されてきたパケットを上層が使用でき

るように整形するパケット整形手段と、整形されたパケットを記憶手段に書き込む書き込み手段とを含み、前記パケット整形手段が、パケットにエラーがあったか否かを示すエラーステータス情報を、パケットの制御情報に付加することを特徴とするデータ転送制御装置。

【請求項 6】 請求項 5 において、整形されたパケットの制御情報を前記記憶手段の制御情報領域に書き込み、整形されたパケットのデータを前記記憶手段のデータ領域に書き込むパケット分離手段を含み、パケットにエラーがあると判断された場合には、前記データ領域に書き込まれた該パケットのデータを無効にすることを特徴とするデータ転送制御装置。

【請求項 7】 バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、各ノードから転送されてきたパケットを上層が使用できるように整形するパケット整形手段と、整形されたパケットを記憶手段に書き込む書き込み手段とを含み、前記パケット整形手段が、自己識別期間に受信したパケットか否かを示す情報を、パケットの制御情報に付加することを特徴とするデータ転送制御装置。

【請求項 8】 バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、各ノードから転送されてきたパケットを上層が使用できるように整形するパケット整形手段と、整形されたパケットを記憶手段に書き込む書き込み手段とを含み、前記パケット整形手段が、各ノードから転送されてきたパケットに対して時系列で後ろ側に所与の情報を付加し、前記書き込み手段が、前記所与の情報を、前記記憶手段においてパケットの制御情報の先頭側に書き込むことを特徴とするデータ転送制御装置。

【請求項 9】 請求項 8 において、前記所与の情報が、パケットの転送元に返したアクノリッジメント情報、データ転送の速度を特定するためのスピードコード情報、バスに接続される全てのノードに対して送信されたパケットか否かを示すブロードキャスト情報、自己識別期間に受信したパケットか否かを示す情報、及びパケットにエラーがあったか否かを示すエラーステータス情報の少なくとも 1 つであることを特徴とするデータ転送制御装置。

【請求項 10】 請求項 1 乃至 9 のいずれかにおいて、前記パケット整形手段と前記記憶手段との間に設けられ

るFIFOを含むことを特徴とするデータ転送制御装置。

【請求項11】 請求項1乃至10のいずれかにおいて、

IEEE1394の規格に準拠したデータ転送を行うことを特徴とするデータ転送制御装置。

【請求項12】 請求項1乃至11のいずれかのデータ転送制御装置と、
前記データ転送制御装置及びバスを介して他のノードから受信したデータに所与の処理を施す装置と、
処理が施されたデータを出力又は記憶するための装置とを含むことを特徴とする電子機器。

【請求項13】 請求項1乃至11のいずれかのデータ転送制御装置と、
前記データ転送制御装置及びバスを介して他のノードに送信するデータに所与の処理を施す装置と、
処理が施されるデータを取り込むための装置とを含むことを特徴とする電子機器。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、データ転送制御装置及びこれを含む電子機器に関する。

【0002】

【背景技術及び発明が解決しようとする課題】近年、IEEE1394と呼ばれるインターフェース規格が脚光を浴びている。このIEEE1394は、次世代のマルチメディアにも対応可能な高速シリアルバスインターフェースを規格化したものである。このIEEE1394によれば、動画像などのリアルタイム性が要求されるデータも扱うことができる。また、IEEE1394のバスには、プリンタ、スキャナ、CD-Rドライブ、ハードディスクドライブなどのコンピュータの周辺機器のみならず、ビデオカメラ、VTR、TVなどの家庭用電化製品も接続できる。このため、電子機器のデジタル化を飛躍的に促進できるものとして期待されている。

【0003】このようなIEEE1394の概要については、例えば「IEEE1394ハイ・パフォーマンス・シリアルバスの概要」(Interface Apr. 1996の1~10頁)、「PC周辺機器用バス規格群総覧」(Interface Jan. 1997の106頁~116頁)、「IEEE1394-1995(FireWire)のリアルタイム転送モードとマルチメディア対応プロトコル」(Interface Jan. 1997の136~146頁)に開示されている。また、IEEE1394に準拠したデータ転送制御装置としては、テキサス・インスツルメンツ社製のTSB12LV31などが知られている。

【0004】しかしながら、このIEEE1394に準拠したデータ転送制御装置には次のような課題があることが判明した。

【0005】即ち、現在のIEEE1394規格によれば最大で400Mbpsの転送速度が実現可能となっている。しかし、現実には、処理のオーバーヘッドの存在に起因して、システム全体の実転送速度はこれよりもかなり低くなっている。つまり、CPU上で動作するファームウェアやアプリケーションソフトが、送信データを準備したり、受信データを取り込んだりするなどの処理に多くの時間を要してしまい、バス上での転送速度が速くても、結局、高速なデータ転送を実現できない。

10 【0006】特に、周辺機器に組み込まれるCPUは、パーソナルコンピュータなどのホストシステムに組み込まれるCPUに比べて処理能力が低い。このため、ファームウェアやアプリケーションソフトの処理のオーバーヘッドの問題は、非常に深刻なものとなる。従って、このようなオーバーヘッドの問題を効果的に解消できる技術が望まれている。

20 【0007】本発明は、以上のような技術的課題に鑑みてなされたものであり、その目的とするところは、ファームウェアやアプリケーションソフトなどの処理のオーバーヘッドを軽減し、小規模なハードウェアで高速なデータ転送を実現できるデータ転送制御装置及びこれを用いられる電子機器を提供することにある。

【0008】

【課題を解決するための手段】上記課題を解決するために本発明は、バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、各ノードから転送されてきたパケットを上層が使用できるように整形するパケット整形手段と、整形されたパケットの制御情報をランダムアクセス可能な記憶手段の制御情報領域に書き込み、整形されたパケットのデータを前記記憶手段のデータ領域に書き込むパケット分離手段とを含み、前記パケット分離手段が、前記データ領域に書き込まれるデータのアドレスを指すデータポインタを前記パケット整形手段に渡し、前記パケット整形手段が、前記パケット分離手段から渡された前記データポインタをパケットの制御情報に付加することを特徴とする。

40 【0009】本発明によれば、パケット整形手段が、パケット分離手段から渡されたデータポインタをパケットの制御情報に付加する。そして、データポインタが付加されたパケットの制御情報が記憶手段の制御情報領域に書き込まれ、パケットのデータがデータ領域に書き込まれる。このように、制御情報を制御情報領域に書き込み、データをデータ領域に書き込むことで、ファームウェアやアプリケーションソフトなどの上層の処理負荷を軽減でき、システム全体の実転送速度を向上できるようになる。また、ファームウェアなどの上層が、データポインタを使用して、パケットの制御情報に対応するデータをデータ領域から容易に読み出すことができるようになる。また、データポインタの付加はパケット整形手段により行われ、パケット分離手段がこれに関与する必要

がないため、パケット分離手段の構成や記憶手段への書き込み処理を簡素化できる。

【0010】また本発明は、前記パケット整形手段が、パケットの少なくとも制御情報とデータを区別するためのタグ情報を生成すると共に、生成された該タグ情報をパケットに関連づけ、前記パケット分離手段が、パケットに関連づけられた前記タグ情報に基づいて、パケットの制御情報を前記制御情報領域に書き込み、パケットのデータを前記データ領域に書き込むことを特徴とする。このようにすることで、簡易なハードウェア構成で、パケットの制御情報を制御情報領域にデータをデータ領域に分離して格納することが可能になる。

【0011】また本発明は、前記パケット分離手段が、パケットの制御情報が書き込まれていると、パケットに関連づけられた前記タグ情報に基づき判断した場合には、前記制御情報領域に書き込まれる制御情報のアドレスを指す制御情報ポインタを更新し、パケットのデータが書き込まれていると、前記タグ情報に基づき判断した場合には、前記データ領域に書き込まれるデータのアドレスを指すデータポインタを更新することを特徴とする。このようにすれば、制御情報ポインタの更新処理とデータポインタの更新処理をタグ情報を利用して切り替え、パケットを制御情報とデータに分離できるようになる。これにより、パケットの分離処理の簡素化を図れる。

【0012】また本発明は、バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、各ノードから転送されてきたパケットを上層が使用できるように整形するパケット整形手段と、整形されたパケットを記憶手段に書き込む書き込み手段とを含み、前記パケット整形手段が、バスに接続される全てのノードに対して送信されたパケットか否かを示すブロードキャスト情報を、パケットの制御情報に付加することを特徴とする。

【0013】本発明によれば、ファームウェアなどの上層は、処理対象となるパケットがブロードキャストされたパケットか否かを、パケットの制御情報に付加されたブロードキャスト情報を調べるだけで知ることができる。このため、他の情報をデコードするなどの処理を省略することが可能になり、ファームウェアなどの上層の処理負担を軽減できる。

【0014】また本発明は、バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、各ノードから転送されてきたパケットを上層が使用できるように整形するパケット整形手段と、整形されたパケットを記憶手段に書き込む書き込み手段とを含み、前記パケット整形手段が、パケットにエラーがあったか否かを示すエラーステータス情報を、パケットの制御情報に付加することを特徴とする。

【0015】本発明によれば、ファームウェアなどの上

層は、処理対象となるパケットにエラーがあるか否かを、パケットの制御情報に付加されたエラーステータス情報を調べるだけで知ることができる。このため、他の情報をデコードするなどの処理を省略することが可能になり、ファームウェアなどの上層の処理負担を軽減できる。

【0016】また本発明は、整形されたパケットの制御情報を前記記憶手段の制御情報領域に書き込み、整形されたパケットのデータを前記記憶手段のデータ領域に書き込むパケット分離手段を含み、パケットにエラーがあると判断された場合には、前記データ領域に書き込まれた該パケットのデータを無効にすることを特徴とする。このようにすることで、データ領域に、正しいデータのみを連続して格納できるようになり、アプリケーションソフトなどの上層がデータ領域から連続して正しいデータを読み出すことが可能になる。これにより、上層とのインターフェースの簡素化を図れるようになる。

【0017】また本発明は、バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、各ノードから転送されてきたパケットを上層が使用できるように整形するパケット整形手段と、整形されたパケットを記憶手段に書き込む書き込み手段とを含み、前記パケット整形手段が、自己識別期間に受信したパケットか否かを示す情報を、パケットの制御情報に付加することを特徴とする。

【0018】本発明によれば、ファームウェアなどの上層は、処理対象となるパケットが自己識別期間に受信したパケットか否かを、パケットの制御情報に付加された情報を調べるだけで簡易に知ることができる。これにより、例えば、複数の自己識別パケットを1つにパッケージングすることで得られたパケットと、それ以外のパケットとの区別などが容易になる。

【0019】また本発明は、バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、各ノードから転送されてきたパケットを上層が使用できるように整形するパケット整形手段と、整形されたパケットを記憶手段に書き込む書き込み手段とを含み、前記パケット整形手段が、各ノードから転送されてきたパケットに対して時系列で後ろ側に所与の情報を付加し、前記書き込み手段が、前記所与の情報を、前記記憶手段においてパケットの制御情報の先頭側に書き込むことを特徴とする。

【0020】本発明によれば、時系列でパケットの後ろ側に付加された情報が、記憶手段においてパケットの制御情報の先頭側に書き込まれる。従って、ファームウェアなどの上層は、他の情報をデコードすることなく、所与の情報を簡易に確認できるようになる。この結果、ファームウェアなどの上層の処理負担を大幅に軽減できる。

【0021】なお本発明では、前記所与の情報は、パケ

ットの転送元に返したアクノリッジメント情報、データ転送の速度を特定するためのスピードコード情報、バスに接続される全てのノードに対して送信されたパケットか否かを示すブロードキャスト情報、自己識別期間に受信したパケットか否かを示す情報、及びパケットにエラーがあったか否かを示すエラーステータス情報の少なくとも1つであることが望ましい。

【0022】また本発明では、前記パケット整形手段と前記記憶手段との間に設けられるFIFOを含むようにしてもよい。

【0023】また本発明では、IEEE1394の規格に準拠したデータ転送を行うことが望ましい。

【0024】また本発明に係る電子機器は、上記のいずれかのデータ転送制御装置と、前記データ転送制御装置及びバスを介して他のノードから受信したデータに所与の処理を施す装置と、処理が施されたデータを出力又は記憶するための装置とを含むことを特徴とする。また本発明に係る電子機器は、上記のいずれかのデータ転送制御装置と、前記データ転送制御装置及びバスを介して他のノードに送信するデータに所与の処理を施す装置と、処理が施されるデータを取り込むための装置とを含むことを特徴とする。

【0025】本発明によれば、他のノードから転送されたデータを電子機器において出力したり記憶したりする処理、電子機器において取り込んだデータを他のノードに転送したりする処理を高速化することが可能になる。また、本発明によれば、データ転送制御装置を小規模化できると共に、データ転送を制御するファームウェアなどの処理負担を軽減できるため、電子機器の低コスト化、小規模化などを図ることも可能になる。

【0026】

【発明の実施の形態】以下、本発明の好適な実施形態について図面を用いて詳細に説明する。

【0027】1. IEEE1394

まず、IEEE1394の概要について簡単に説明する。

【0028】1. 1 データ転送速度、接続トポロジー IEEE1394 (IEEE1394-1995、IEEE1394. A) では100~400Mbpsの高速なデータ転送が可能となっている (IEEE1394. Bでは800~3200Mbps)。また、転送速度が異なるノードをバスに接続することも許される。

【0029】各ノードはツリー状に接続されており、1つのバスに最大で63個のノードが接続可能になっている。なお、バスブリッジを利用すれば約64000個のノードを接続することも可能である。

【0030】電源が投入されたり、途中でデバイスの抜き差しが発生すると、バスリセットが発生し、接続トポロジーに関する情報が全てクリアされる。そして、バスリセット後、ツリー識別 (ルートノードの決定)、自己

識別が行われる。その後、アイソクロナスリソースマネージャ、サイクルマスタ、バスマネージャ等の管理ノードが決定される。そして、通常のパケット転送が開始される。

【0031】1. 2 転送方式 IEEE1394では、パケットの転送方式として、信頼性が要求されるデータの転送に好適な非同期転送と、リアルタイム性が要求される動画や音声などのデータの転送に好適なアイソクロナス転送が用意されている。

10 【0032】図1 (A) に、非同期サブアクションの例を示す。1つのサブアクションは、調停、パケット転送、アクノリッジメントからなる。即ち、データ転送に先立って、まず、バスの使用権に関する調停が行われる。そしてソース (転送元) ノードからデスティネーション (転送先) ノードにパケットが転送される。このパケットのヘッダにはソースIDとデスティネーションIDが含まれる。デスティネーションノードは、このデスティネーションIDを読んで、自ノード宛のパケットか否かを判断する。デスティネーションノードは、パケットを受け取ると、ソースノードにアクノリッジメント (ACK) のパケットを返す。

【0033】パケット転送とACKの間にはアクノリッジギャップが存在する。また、1つのサブアクションと次のサブアクションの間にはサブアクションギャップが存在する。そして、サブアクションギャップに相当する一定のバス・アイドル時間が経過しないと、次のサブアクションの調停を開始できない。これによりサブアクション相互の衝突が回避される。

30 【0034】図1 (B) に、アイソクロナスサブアクションの例を示す。アイソクロナス転送はブロードキャスト (バスに接続される全てのノードに転送) で実行されるため、パケット受信時にACKは返送されない。また、アイソクロナス転送では、ノードIDではなくチャンネル番号を使用してパケット転送が行われる。なお、サブアクション間にはアイソクロナスギャップが存在する。

【0035】図1 (C) に、データ転送時のバスの様子を示す。アイソクロナス転送は、サイクルマスタが一定周期毎にサイクルスタートパケットを発生することで開始する。これにより、1つのチャンネル当たり、125μs毎に少なくとも1つのパケットを転送できるようになる。この結果、動画や音声などのリアルタイム性が要求されるデータの転送が可能になる。

【0036】非同期転送はアイソクロナス転送の合間に行われる。即ち、アイソクロナス転送の方が非同期転送よりも優先順位が高くなっている。これは、図1 (C) に示すように、アイソクロナスギャップの時間を、非同期転送のサブアクションギャップの時間よりも短くすることで実現される。

50 【0037】1. 3 ツリー識別

ツリー識別はバスリセットの後に行われる。ツリー識別によりノード間の親子関係やルートノードが決定される。

【0038】まず、リーフノード（1つのノードにしか接続されていないノード）が、隣接するノードに、ペアレントノティファイを送る。例えば図2（A）のようにノードA、B、C、D、Eが接続されている場合には、ノードAからノードBに、ノードD及びEからノードCに、ペアレントノティファイ（PN）が送られる。

【0039】ペアレントノティファイを受け取ったノードは、送り元のノードを自身の子と認知する。そして、チャイルドノティファイをそのノードに送る。例えば図2（A）では、ノードBからノードAに、ノードCからノードD及びEにチャイルドノティファイ（CN）が送られる。これによってノードB、A間、ノードC、D間、ノードC、E間の親子関係が決定する。

【0040】ノードB、Cの親子関係は、どちらが先にペアレントノティファイを送ったかで決定される。例えば図2（B）のように、ノードCの方が先にペアレントノティファイを送ると、ノードBが親になりノードCが子になる。

【0041】ポートの接続先の全てのノードが自身の子となるノードがルートになる。図2（B）では、ノードBがルートになる。なお、IEEE1394では、全てのノードがルートになる可能性がある。

【0042】1. 4 自己識別

ツリー識別の後、自己識別が行われる。自己識別においては、接続トポロジーにおいてルートノードから遠いノードから順にセルフIDパケットが転送される。

【0043】より具体的には、例えば図3において、まず、ルートノードBのポート1（番号の小さいポート）に接続されるノードAが、セルフIDパケット（自己識別パケット）を全てのノードにブロードキャストする。

【0044】次に、ルートノードBのポート2（番号が大きいポート）に接続されるノードCが選択され、このノードCのポート1（番号の小さいポート）に接続されるノードDがセルフIDパケットをブロードキャストする。次に、ノードCのポート2（番号の大きいポート）に接続されるノードEがセルフIDパケットをブロードキャストし、その後、ノードCがブロードキャストする。最後に、ルートであるノードBがセルフIDパケットをブロードキャストし、自己識別が完了する。

【0045】セルフIDパケットには各ノードのIDが含まれる。ブロードキャストを行う時点で他のノードから受け取ったセルフIDパケットの個数が、この各ノードのIDとなる。例えば図3では、ノードAがブロードキャストを行う時点では、どのノードもセルフIDパケットを発していないため、ノードAのIDは0になる。ノードAは、このID=0をセルフIDパケットに含

せてブロードキャストする。また、ノードDがブロードキャストを行う時点では、ノードAのみがセルフIDパケットを発している。このため、ノードDのIDは1になる。同様に、ノードE、C、BのIDは、各々、2、3、4になる。

【0046】図4（A）にセルフIDパケットのフォーマットを示す。同図に示すようにセルフIDパケットには各ノードの基本情報が含まれる。具体的には、各ノードのID（PHY_ID）、リンク層がアクティブか否か（L）、ギャップカウント（gap_cnt）、転送速度（sp）、アイソクロナスリソースマネージャになれる能力を有するか否か（C）、電力状態（pwr）、ポートの状態（p0、p1、p2）などに関する情報が含まれる。

【0047】なお、図4（B）に、ノードのポート数が4個以上の場合に使用されるセルフIDパケット#1、#2、#3のフォーマットを示す。ポート数が4～11個の場合にはセルフIDパケット#0（図4（A））及び#1が、12～19個の場合にはセルフIDパケット#0、#1及び#2が、20～27個の場合にはセルフIDパケット#0、#1、#2及び#3が使用されることになる。

【0048】また、図4（C）、（D）に、セルフIDパケットと同様に、物理層のパケット（PHYパケット）であるリンクオンパケット、PHY構成パケットのフォーマットを示す。

【0049】1. 5 アイソクロナスリソースマネージャ

アイソクロナスリソースマネージャ（IRM）は以下の管理機能を有する。

【0050】第1に、アイソクロナス転送に必要な種々のリソースを提供する。例えば、チャンネル番号レジスタや帯域幅レジスタを提供する。第2に、バスマネージャのIDを示すレジスタを提供する。第3に、バスマネージャがない場合に、簡易的なバスマネージャとなる役割を有する。

【0051】IRMになれる能力を有し（アイソクロナスリソースを管理する能力を有し）、且つ、動作状態になっている（リンク層がアクティブになっている）ノードの中で（IRMになれる資格を有するノードの中で）、ルートに最も近い（IDが最も大きい）ノードがIRMになる。より、具体的には、図4（A）のセルフIDパケットにおいて、IRMになれる能力を有するか否かを示すC（CONTENDER）ビットと、リンク層がアクティブか否かを示すL（LINK_ACTIVE）ビットが共に1になっているノードの中で、ルートに一番近いノード（PHY_IDが一番大きいノード）がIRMになる。例えば、ルートノードのセルフIDパケットのCビットとLビットが1の場合には、ルートノードがIRMになる。

【0052】1. 6 サイクルマスタ、バスマネージャ
サイクルマスタは、図1 (C) に示すサイクルスタート
パケットを送信する役割を有し、ルートノードがサイク
ルマスタになる。

【0053】バスマネージャは、トポロジーマップ (各
ノードの接続状態) の作成、スピードマップの作成、バ
スの電力管理、サイクルマスタの決定、ギャップカウ
ントの最適化などの仕事を行う。

【0054】1. 7 プロトコル構成

図5を用いて、IEEE1394のプロトコル構成 (層
構造) について説明する。

【0055】IEEE1394のプロトコルは、物理
層、リンク層、トランザクション層により構成される。
また、シリアルバスマネージメントは、物理層、リン
ク層、トランザクション層をモニターしたり制御したり
するものであり、ノードの制御やバスのリソース管理の
ための種々の機能を提供する。

【0056】物理層は、リンク層により使用されるロジ
カルシンボルを電気信号に変換したり、バスの調停を行
ったり、バスの物理的インターフェースを定義する。

【0057】リンク層は、アドレッシング、データチェ
ック、データフレーミング、サイクル制御などを提供す
る。

【0058】トランザクション層は、リード、ライト、
ロックなどのトランザクションを行うためのプロトコル
を定義する。

【0059】物理層及びリンク層は、通常、データ転送
制御装置 (インターフェースチップ) などのハードウェ
アにより実現される。また、トランザクション層は、C
PU上で動作するファームウェアや、ハードウェアによ
り実現される。

【0060】2. 全体構成

次に、本実施形態の全体構成について図6を用いて説明
する。

【0061】図6において、PHYインターフェース1
0は、物理層のプロトコルを実現するPHYチップとの
インターフェースを行う回路である。

【0062】リンクコア20 (リンク手段) は、リンク
層のプロトコルやトランザクション層のプロトコルの一
部を実現する回路であり、ノード間でのパケット転送の
ための各種サービスを提供する。レジスタ22は、これ
らのプロトコルを実現したりリンクコア20を制御する
ためのレジスタである。

【0063】FIFO (ATF) 30、FIFO (IT
F) 32、FIFO (RF) 34は、各々、非同期送信用、
アイソクロナス送信用、受信用のFIFOであり、
例えばレジスタや半導体メモリなどのハードウェアによ
り構成される。本実施形態では、これらのFIFO3
0、32、34の段数は非常に少ない。例えば1つのF
IFOの段数は、好ましくは3段以下であり、更に好ま

しくは2段以下となる。

【0064】DMAC40、42、44は、各々、AT
F、ITF、RF用のDMAコントローラである。これ
らのDMAC40、42、44を用いることで、CPU
66に介入されることなく、RAM80とリンクコア2
0との間でのデータ転送が可能になる。なお、レジスタ
46は、DMAC40、42、44などを制御するレジ
スタである。

【0065】ポートインターフェース50は、アプリケ
ーション層のデバイス (例えばプリンタの印字処理を行
うデバイス) とのインターフェースを行う回路である。
本実施形態では、このポートインターフェース50を用
いて、例えば8ビットのデータ転送が可能になってい
る。

【0066】FIFO (PF) 52は、アプリケーシ
ョン層のデバイスとの間でのデータ転送のためのFIFO
であり、DMAC54は、PF用のDMAコントローラ
である。レジスタ56は、ポートインターフェース50
やDMAC54を制御するレジスタである。

【0067】CPUインターフェース60は、データ転
送制御装置をコントロールするCPU66とのインター
フェースを行う回路である。CPUインターフェース6
0は、アドレスデコーダ62、データ同期化回路63、
割り込みコントローラ64を含む。クロック制御回路6
8は、本実施形態で使用されるクロックを制御するも
のであり、PHYチップから送られてくるSCLKや、マ
スタークロックであるHCLKが入力される。

【0068】バッファマネージャ70は、RAM80と
のインターフェースを管理する回路である。バッファマ
ネージャ70は、バッファマネージャの制御のためのレ
ジスタ72、RAM80へのバス接続を調停する調停回
路74、各種の制御信号を生成するシーケンサ76を含
む。

【0069】RAM80は、ランダムアクセス可能なパ
ケット記憶手段として機能するものであり、その機能は
例えばSRAM、DRAMなどにより実現される。そし
て、本実施形態では、図7に示すように、このRAM8
0がヘッダ領域 (広義には制御情報領域) とデータ領域
に分離されている。そして、パケットのヘッダ (広義に
は制御情報) は図7のヘッダ領域に格納され、パケット
のデータはデータ領域に格納される。

【0070】なおRAM80は、本実施形態のデータ転
送制御装置に内蔵させることが特に望ましいが、その一
部又は全部を外付けにすることも可能である。

【0071】バス90 (或いはバス92、94) は、ア
プリケーションに接続されるものである (第1のバス)
。またバス96 (或いはバス98) はデータ転送制
御装置をコントロールするためのものであり、データ転
送制御装置をコントロールするデバイス (例えばCP
U) に電氣的に接続される (第2のバス)。またバス1

00（或いはバス102、104、105、106、107、108、109）は、物理層のデバイス（例えばPHYチップ）に電氣的に接続されるものである（第3のバス）。また、バス110は、ランダムアクセス可能な記憶手段であるRAMに電氣的に接続されるものである（第4のバス）。

【0072】バッファマネージャ70の調停回路74は、DMAC40、DMAC42、DMAC44、CPUインターフェース60、DMAC54からのバスアクセス要求の調停を行う。そして、この調停結果に基づいて、各々、バス105、107、109、98、94のいずれかと、RAM80のバス110との間にデータの経路が確立される（第1、第2、第3のバスのいずれかと第4のバスとの間にデータ経路が確立される）。

【0073】本実施形態の1つの特徴は、ランダムアクセスが可能でありパケットを格納するRAM80を設けると共に、互いに分離されるバス90、96、100と、これらのバスをRAM80のバス110に接続するための調停回路74とを設けた点にある。

【0074】例えば図8に、本実施形態と構成の異なるデータ転送制御装置の例を示す。このデータ転送制御装置では、リンクコア902は、PHYインターフェース900、バス922を介してPHYチップと接続される。また、リンクコア902は、FIFO904、906、908、CPUインターフェース910、バス920を介してCPU912に接続される。そして、CPU912は、バス924を介して、CPUにローカルなメモリであるRAM914に接続される。

【0075】なお、FIFO904、906、908は、図6のFIFO30、32、34と異なり、非常に段数の多いものとなる（例えば1つのFIFOが16段程度）。

【0076】図8の構成のデータ転送制御装置を用いた場合のデータ転送の手法について図9を用いて説明する。PHYチップ930を介して他のノードから送られてきた受信パケットは、バス922、データ転送制御装置932、バス920を介してCPU912が受け取る。そして、CPU912は、受け取った受信パケットをバス924を介してRAM914に書き込む。そして、CPU912は、受信パケットをアプリケーション層が使用できるように加工し、バス926を介してアプリケーション層のデバイス934に転送する。

【0077】一方、アプリケーション層のデバイス934からのデータを転送する場合には、CPU912は、このデータをRAM914に書き込む。そして、RAM914のデータにヘッダを付加することでIEEE1394に準拠したパケットを生成する。そして生成されたパケットは、データ転送制御装置932、PHYチップ930などを介して他のノードに送信される。

【0078】しかしながら、このようなデータ転送手法

によると、CPU912の処理負荷が非常に重くなる。従って、ノード間を接続するシリアルバスの転送速度が高速になっても、CPU912の処理のオーバーヘッドなどに起因して、システム全体の実転送速度は低くなり、結局、高速なデータ転送を実現できない。

【0079】このような問題を解決する1つの手法として、図10に示すように、データ転送制御装置932とRAM914との間でのデータ転送や、RAM914とアプリケーション層のデバイス934との間でのデータ転送を、ハードウェアDMAにより実現する手法も考えられる。

【0080】しかしながら、この手法では、CPUバス928が、データ転送制御装置932、RAM914間でのデータ転送、RAM914、CPU912間でのデータ転送、RAM914、アプリケーション層デバイス934間でのデータ転送に使用されることになる。従って、システム全体のデータ転送の高速化を図ろうとすると、CPUバス928としてPCIバスのような高速なバスを使用しなければならなくなり、これは、データ転送制御装置を使用する電子機器の高コスト化を招く。

【0081】これに対して、本実施形態では図11に示すように、データ転送制御装置120、アプリケーション層デバイス124間のバス90と、CPUバス96と、データ転送制御装置120、RAM80間のバス110とが互いに分離されている。従って、CPUバス96をデータ転送の制御のみに使用できるようになる。また、バス90を占有して、データ転送制御装置120、アプリケーション層デバイス124間でデータ転送を行うことができるようになる。例えば、データ転送制御装置120が組み込まれる電子機器がプリンタである場合には、バス90を占有して印字データを転送できるようになる。この結果、CPU66の処理負荷を軽減でき、システム全体の実転送速度を高めることができる。またCPU66として安価なものを採用できると共に、CPUバス96として高速なバスを使用する必要性がなくなる。このため、電子機器の低コスト化、小規模化を図れるようになる。

【0082】3. 受信パケットフォーマット

3. 1 本実施形態の特徴

本実施形態では、図7に示すように、RAM80の記憶領域を、パケットのヘッダ（広義には制御情報）が格納されるヘッダ領域（広義には制御情報領域）とパケットのデータが格納されるデータ領域に分離している。

【0083】例えば図8の比較例では、CPU912は、受信した順序でFIFO908から受信パケットを受け取らなければならない。従って、受信パケットをヘッダとデータに分離する処理を実現しようとする、FIFO908から読み出した受信パケットをCPU912がローカルメモリであるRAM914に一旦書き込み、そのRAM914からCPU912が受信パケット

を読み出してヘッダとデータに分離する必要がある。従って、CPU 912の処理負荷が非常に重くなり、これは結局、システム全体の実転送速度の低下という事態を招く。

【0084】これに対して、図6の本実施形態では、RAM 80の記憶領域がヘッダ領域とデータ領域に分離される。より具体的には、図12に示すように、各受信パケットのヘッダとデータがハードウェアにより分離され、ヘッダがヘッダ領域に、データがデータ領域に格納される。従って、CPU 66の処理負荷が図8に比べて非常に軽くなり、システム全体の実転送速度を向上できる。また、CPU 66として安価なものを採用できると共にCPU 66に接続されるバスも低速なもので済むため、データ転送制御装置や電子機器の小規模化、低コスト化を図れるようになる。

【0085】また、本実施形態によれば、ヘッダはヘッダ領域にまとめられて格納され、データもデータ領域にまとめられて格納される。従って、ヘッダやデータの読み出し処理や書き込み処理の簡易化を図ることができ、処理のオーバーヘッドを軽減化できる。例えば、図11の手法でデータ転送を行う場合には、CPU 66は、CPUバス96を介してヘッダ領域にのみアクセスしヘッダを読み出したり書き込んだりするだけでデータ転送を制御できるようになる。また、アプリケーション層のデバイス124は、データ領域のデータを連続してバス90を介して読み出したり、データを連続してデータ領域に書き込んだりすることが可能になる。

【0086】そして、本実施形態では、図13に示すように、ヘッダ領域に格納される各ヘッダと、データ領域に格納される各データとを、ヘッダに含ませたデータポインタにより対応させている。このデータポインタの付加は以下のようにして実現する。

【0087】即ち図14に示すように、本実施形態では、リンクコア20がパケット整形回路160を含み、DMAC 44（広義には書き込み手段）がパケット分離回路180を含む。ここでパケット整形回路160は、各ノードから転送されてきたパケットを上層（例えばトランザクション層、アプリケーション層）が使用できるように整形する。また、パケット分離回路180は、パケットを分離し、パケットのヘッダやトレーラー（フッター）をRAM 80のヘッダ領域に書き込み、パケットのデータをRAM 80のデータ領域に書き込むための処理を行う。

【0088】パケット分離回路180は、データ領域に書き込まれるデータのアドレス（先頭アドレス）を指すデータポインタをパケット整形回路160に渡す。データポインタを渡されたパケット整形回路160は、そのデータポインタをパケットのヘッダに付加する（埋め込む）。そして、データポインタが付加されたヘッダは、図13に示すようにヘッダ領域に格納され、そのデータ

ポインタが指すデータは、データ領域に格納される。この場合、ヘッダやデータを書き込むアドレスWADRはパケット分離回路180が発生することになる。

【0089】以上のようにすることで、ヘッダ領域からヘッダを読み出したファームウェアなどが、そのヘッダに対応するデータのデータ領域での格納アドレスを容易に知ることができるようになる。また、データポインタの付加は、パケット整形回路160により行われ、DMAC 44（パケット分離回路180）はこれに関与する必要がない。従って、DMAC 44がRAM 80へのデータ書き込み処理に専念できるようになり、DMAC 44の回路構成や処理を簡素化できるようになる。

【0090】また、本実施形態では、図15に示すように、受信パケットのスタート（ヘッダの最初）、ヘッダ、データ、トレーラーを区別するためのTAGを生成し、このTAGを受信パケットに関連づけている。より具体的には、図14において、リンクコア20が、受信パケットのスタート（ヘッダの最初）、ヘッダ、データ、トレーラーをRDとしてFIFO 34に転送する際に、図15に示す2ビットのTAGも同時にFIFO 34に出力する。そして、本実施形態では、受信パケットに関連づけられたこのTAGを利用して、図12に示すように、受信パケットをヘッダとデータに分離しヘッダ領域とデータ領域に格納している。より具体的には、図14において、DMAC 44のパケット分離回路180が、受信パケットと共にFIFO 34から出力されるDTAG（＝TAG）を判別し、この判別結果に基づいて受信パケットをヘッダとデータに分離し、RAM 80に書き込んでいる。なお、TAGは、少なくともヘッダとデータを区別できるものであればよい。

【0091】例えば、TAGを利用せずに受信パケットのヘッダとデータを分離する手法として、ヘッダに含まれるtcode（トランザクションコード）を利用する手法が考えられる。即ち、tcodeをデコードして、ヘッダのサイズを調べ、そのサイズの分だけ受信パケットをヘッダ領域に格納し、残りをデータ領域に格納する。

【0092】しかしながら、この手法では、tcodeをデコードする回路が必要となり、回路が大規模化してしまう。

【0093】これに対して、TAGを利用すれば、DMAC 44は、TAGを見るだけで受信パケットをヘッダとデータに分離できる。従って、回路をそれほど大規模化することなく、簡易な処理で受信パケットのヘッダとデータを分離できるようになる。

【0094】なお、本実施形態では図16に示すように、RAM 80のヘッダ領域を、受信用、送信用の領域に分離することが望ましい。また、データ領域を、受信用、送信用の領域に分離したり、アイソクロナス転送用、非同期転送用の領域に分離するようにしてもよい。

また、ヘッダ領域とデータ領域の他に、これらの領域と分離されたCPU66のワーク領域を設けることが望ましい。

【0095】更に、RAM80の記憶領域が複数の領域に分離される場合には、各領域のサイズを可変に制御することが望ましい。より具体的には図16に示すように、各領域の境界のアドレスを指すポインタP1~P6を可変に制御する。このようにすれば、アプリケーションに応じた最適な領域分割を実現できる。この場合、RAM80の各領域のサイズは、電源投入後もダイナミックに可変制御できることが望ましい。このようにすれば、受信処理の際には受信用の領域を広くし、送信処理の際には送信用の領域を広くすることが可能になり、限られたリソースを有効利用できるようになる。

【0096】また、本実施形態では、図17(A)に示すように、バスに接続される全てのノードに対して送信されたパケットか否かを示すBC(広義にはブロードキャスト情報)を、パケット整形の際に、パケットの例えばトレイラー(広義には制御情報)に付加する。

【0097】即ち、IEEE1394の非同期転送では、転送先のノードを指定するためにパケットに対して16ビットのデスティネーションID(DestID)を含ませる。そして、図17(B)に示すように、リンクコア(リンク層)は、他ノード宛のパケットについては上層のファームウェア(トランザクション層)に伝えないようにする。このようにすることで、16ビットのDestIDをデコードするなどの無駄な処理をファームウェアが行うのを防止できる。

【0098】しかしながら、パケットに含まれるDestIDが0x3Fである場合には、そのパケットは全てのノードにブロードキャストされたものになる。従って、リンクコアは、そのパケットを上層のファームウェアに伝えることになる。このため、結局、ファームウェアは、そのパケットのDestIDをデコードして、ブロードキャストされたものか否かを判断しなければならない。

【0099】そこで、本実施形態では、図17(C)に示すように、DestIDが0x3Fである場合には、ブロードキャスト情報であるBCを1にする。一方、図17(D)に示すように、DestIDが0x3Fでない場合には、BCを0にする。このようにすれば、ファームウェアは、1ビットのBCの値を調べるだけで、そのパケットがブロードキャストされたものか否かを判断できるようになる。即ち、ファームウェアは、16ビットのDestIDをデコードしなくても、そのパケットがブロードキャストされたものか否かを判断できるようになる。従って、ファームウェアはDestIDをデコードしなくてよくなり、ファームウェアの処理負担を軽減できる。この結果、システム全体の実転送速度を向上できるようになる。

【0100】また、本実施形態では、図18(A)に示すように、パケットの例えばヘッダーにエラーがあったか否かを示すHCE(広義にはエラーステータス情報)を、パケット整形の際に、パケットの例えばトレイラー(広義には制御情報)に付加する。

【0101】このようにすれば、ファームウェアは、1ビットのHCEの値を調べるだけで、そのパケットにエラーがあったか否かを判断できるようになる。即ち、ファームウェアは、32ビットのヘッダCRCをデコードしなくても、そのパケットにエラーがあったか否かを判断できるようになる。この結果、ファームウェアの処理負担を軽減でき、システム全体の実転送速度を向上できるようになる。

【0102】なお、パケットにエラーがあると判断された場合には、データ領域に書き込まれた当該パケットのデータを無効にすることが望ましい。より具体的には、図18(B)において、データ3を含むパケットにエラーがあると判断された場合には(HCE=1)、データポインタDPを、データ3が格納されている領域の先頭アドレスに戻す。このようにすることで、データ領域でのデータの連続性を維持できる。即ち、データ領域に、正しいデータのみを連続して格納できるようになり、アプリケーションソフトがデータ領域から連続して正しいデータを読み出すことが可能になる。また、パケットにエラーがあったか否かをファームウェアがHCEにより確認し、ファームウェアが相手ノードに対してパケットの再送信を要求できるようになる。このように本実施形態によれば、ファームウェアやアプリケーションソフトの処理負担を軽減でき、システム全体の実転送速度を向上できるようになる。

【0103】なお、エラーステータス情報としては、ヘッダCRCエラーの他にも、データCRCエラーなどを考えることもできる。

【0104】さて、図19に示すように、セルフID期間においては、バスに接続される全てのノードからセルフIDパケットが送られてくる。そして、各セルフIDパケットの第1クワドレットは、ノードIDなどの基本情報が含まれるデータ本体(ボディ)になっており、第2クワドレットは、第1クワドレットの反転であるパリティ(エラーチェック情報)になっている(詳しくは、図4(A)参照)。

【0105】例えば図8の構成のデータ転送制御装置では、これらの一連のセルフIDパケットを図20(A)に示すように整形する。即ち、物理層のパケットであるセルフIDパケットに対して、上層(トランザクション層やアプリケーション層)にインターフェースするためのヘッダを付加するというパケット整形が行われる。そして、整形されたパケットは、CPU912のローカルメモリであるRAM914に格納される。

【0106】しかしながら、セルフIDパケットは、ノ

ード数に応じた個数だけ必要であり、バスに63個のノードが接続されると、例えば最小で63個のセルフIDパケットを保持する必要がある。即ち、セルフIDパケットの個数は、通常、非常に多い。従って、これらのセルフIDパケットにヘッダを付加し、そのままRAMに格納すると、RAMの空き容量が圧迫されてしまい、各ノードのリソースを有効利用できなくなってしまう。また、それぞれのセルフIDパケットを受け取る毎に、ファームウェアは必要な処理を行わなければならない。従ってセルフIDパケットの個数が増加すると、ファームウェアの処理負担が増加する。更に、セルフID期間が終了した後、即ちバスリセットが発生してから最初のアービトレーション・リセット・ギャップが発生した後は、その受け取ったセルフIDパケットの整形作業が必要になる。従って、セルフIDパケットの個数が増加すると、この整形作業の処理負担も増加する。

【0107】そこで、本実施形態では図20(B)に示すように、転送されてくる一連のパケットを、ヘッダ(広義には制御情報)と、一連のセルフIDパケットからなるデータとによりフレームが構成されるパケットに整形している。つまり、一連のセルフIDパケットを1つにパッケージングし、このパッケージングされたものに1つのヘッダを付加し、上層にインターフェースしている。

【0108】このようにすることで、本実施形態では、パケット記憶手段であるRAM(図6のRAM80)の空き容量が圧迫される事態を効果的に解消することに成功している。また、本実施形態では、セルフIDパケットがN個あった場合にも、整形後のパケットは1個になる。従って、整形後のパケットをRAMから読み出すためには、1回のパケット読み出しを行えば済むようになる。従って、ファームウェアなどの処理負担を格段に軽減できる。

【0109】更に本実施形態では、図20(B)に示すように、セルフID期間に受信したパケットか否かを示すステータス情報であるBRを、パケットのトレイラー(広義には制御情報)に付加している。

【0110】即ち、図4(A)～(D)に示すように、物理層が扱うPHYパケットには、セルフIDパケットの他にリンクオンパケットとPHY構成パケットがある。そして本実施形態では、セルフIDパケットと他のPHYパケット(リンクオンパケット、PHY構成パケット)とをファームウェアなどが容易に区別できることが望ましい。セルフIDパケットにはデータポインタなどが付加されており、セルフIDパケットと他のPHYパケットとではファームウェアの取り扱いが異なったものになるからである。

【0111】ところが、パケットの種類を区別するための情報としてIEEE1394において規格化されているt c o d eを用いても、処理対象となるパケットが、

PHYパケットの中のどのパケットなのかを区別できない。

【0112】本実施形態によれば、図20(B)に示すようにパケットのトレイラーに、セルフID期間に受信したパケットか否かを示すBRが付加される。このため、セルフIDパケットと他のPHYパケットとをファームウェア等が容易に区別できるようになり、処理負担を軽減できる。

【0113】なお、本実施形態では、セルフIDパケットの第2クワドレットのパリティを削除し、このパリティが削除されたセルフIDパケット(セルフIDパケットのデータ本体)の並びにより、整形後のパケットのデータ部分を構成するようにすることが望ましい。このようにすることで、整形後のパケットのサイズを、パリティを削除しない場合に比べて半分程度にすることができ、セルフIDパケットの記憶に必要なRAMの使用容量を更に節約できる。

【0114】さて、図1(A)で説明したように、非同期転送でデスティネーションノードがパケットを受信すると、デスティネーション(転送先)ノードはソース(転送元)ノードに対してACKを返す。その一方で、デスティネーションノードのリンクコアは、受信したパケットに対して、ソースノードに返したACKを付加する。これにより、ソースノードに返したACKをファームウェアなどの上層に伝えることができる。より具体的には、図14のパケット整形回路160が、図21(A)に示すようにトレイラーにACKを付加する。このACKは4ビットのコードであり、このコードにより、アクノリッジコンプリート、アクノリッジペンディング、アクノリッジビジー、アクノリッジエラーなどの状態を知ることができる。

【0115】しかしながら、図21(A)では、パケット整形の際に、時系列で後ろ側にあるトレイラーにACKが付加され、RAMにおいても、ヘッダの後ろ側にACKが書き込まれる。従って、ファームウェアがACKを確認する場合には以下のような処理を行わなければならない。即ち、まずファームウェアは、ヘッダの先頭側にある4ビットのt c o d eをデコードし、ヘッダのサイズを調べる。そして、そのヘッダのサイズに基づきACK(トレイラー)が格納されるアドレスを算出し、ACKを読み出す。従って、ファームウェアがACKを確認するための処理は負担が重く、システム全体の実転送速度の向上の妨げとなる。特に、アクノリッジエラーの場合などは、ACKを確認するために行った処理が全て無駄なものになってしまう。

【0116】そこで、本実施形態では、図21(B)に示すように、パケット整形において時系列でパケットの後ろ側に付加されたACKなどの所与の情報を、RAMにおいて、パケットのヘッダの先頭側(例えば先頭の第1クワドレット)に書き込むようにしてもよい。このよ

うにすれば、ファームウェアは、`tcode`をデコードすることなく、ACKなどの所与の情報を確認できるようになる。ACK等がヘッダの先頭側に格納されており、ヘッダのサイズなどを`tcode`で調べる必要がないからである。

【0117】なお、ヘッダ長（ヘッダのサイズ）を固定にし、ACKなどの所与の情報をヘッダ領域に書き込むようにしてもよい。このようにヘッダ長を固定にすれば、ヘッダ領域中の任意の固定長ヘッダの任意のクワドレットにACKなどの所与の情報を格納したとしても、ファームウェアは、上記所与の情報に簡易にアクセスできるようになるからである。また、この場合に、ヘッダ領域中の任意の固定長ヘッダの先頭クワドレット又は最終クワドレットに、上記所与の情報を格納することが更に望ましい。このようにすれば、固定長ヘッダの先頭又は最終クワドレット以外に上記所与の情報を格納する場合に比べて、ヘッダ内のデータの連続性を維持できるからである。

【0118】また、上記所与の情報としては、ACK（ソースノードに返したアクノリジメント情報）以外にも、`spd`（データ転送の速度を特定するためのスピードコード情報）、`BC`（バスに接続される全てのノードに対して送信されたパケットか否かを示すブロードキャスト情報）、`BR`（セルフID期間に受信したパケットか否かを示す情報）、`HCE`（パケットにエラーがあったか否かを示すエラーステータス情報）など、種々のものを考えることができる。

【0119】3. 2 構成

図22に、リンクコア20、FIFO34、DMAC44の詳細な構成の一例を示す。

【0120】リンクコア20は、バス監視回路130、直列・並列変換回路132、パケット整形回路160を含む。そして、パケット整形回路160は、パケット診断回路142、シーケンサ167、バッファ168、セクタ170を含み、パケット診断回路142は、TAG生成回路162、ヘッダ&トレイラー生成回路164、エラーチェック回路166を含む。

【0121】ここで、バス監視回路130は、PHYインターフェース10を介してPHYチップに接続される8ビット幅のデータバスD、2ビット幅のコントロールバスCTLを監視する回路である。

【0122】直列・並列変換回路132は、データバスDのデータを32ビットのデータに変換する回路である。例えば、転送速度が400Mbpsの場合には8ビットのデータが32ビットのデータに、200Mbpsの場合には4ビットのデータが32ビットのデータに、100Mbpsの場合には2ビットのデータが32ビットのデータに変換される。

【0123】パケット診断回路142は、パケットの診断を行う回路である。TAG生成回路162は、ヘッ

ダ、データ、トレイラーなどを区別するためのTAGを生成する回路であり、ヘッダ&トレイラー生成回路164は、ヘッダ及びトレイラー（フッター）を生成する回路である。また、エラーチェック回路166は、パケットに含まれるパリティなどのエラーチェック情報をチェックしてエラーを検出する回路である。

【0124】シーケンサ167は各種の制御信号を生成するものである。バッファ168、セクタ170は、直列・並列変換回路132からのDI、パケット診断回路142からのヘッダ及びトレイラー、DMAC44からのデータポイントのいずれかを、パケット診断回路142からの信号SELにより選択するためのものである。

【0125】FIFO34は、リンクコア20からの出力データであるRDの位相と、RAM80への書き込みデータであるWDATAの位相とを調整するためのバッファとして機能するものであり、FIFO状態判断回路35を含む。FIFO状態判断回路35は、FIFOが空になると、EMPTYをアクティブにし、FIFOがフルになると、FULLをアクティブにする。

【0126】DMAC44（書き込み手段）は、パケット分離回路180、アクセス要求実行回路190、アクセス要求発生回路192を含む。

【0127】パケット分離回路180は、パケット整形回路160により整形されたパケットを分離して、ヘッダ及びトレイラーをRAM80のヘッダ領域に、データをデータ領域に書き込むための回路である（図12参照）。パケット分離回路180は、TAG判別回路182、ポイント更新回路184、アドレス発生回路188を含む。

【0128】TAG判別回路182は、TAG生成回路162により生成されたTAG（DTAG）を判別する回路である。

【0129】ポイント更新回路184は、TAG判別回路182の出力を受け、RAM80にヘッダやデータを書き込むためのヘッダポイント（広義には制御情報ポイント）やデータポイントを更新するための回路である。

【0130】アドレス発生回路188は、ポイント更新回路184の出力を受け、RAM80への書き込みアドレスWADRを発生する回路である。

【0131】アクセス要求実行回路190は、リンクコア20からのアクセス要求を実行するための回路である。アクセス要求実行回路190は、FIFO状態判断回路35からのFULLがアクティブになると、FULLをアクティブにする。パケット整形回路160内のシーケンサ167は、FULLがアクティブでないことを条件に、RD（RxData）のストローブ信号であるRDSをアクティブにする。

【0132】なおRFAILは、受信における失敗を、シーケンサ167がアクセス要求実行回路190に対し

て知らせるための信号である。

【0133】アクセス要求発生回路192は、RAM80へのアクセス要求を発生するための回路である。アクセス要求発生回路192は、バッファマネージャ70からの書き込みアクノリッジメントであるWACKやFIFO状態判断回路35からのEMPTYを受け、書き込み要求であるWREQをバッファマネージャ70に出力する。

【0134】3.3 受信側の動作

次に、本実施形態の動作の詳細について図23のタイミング波形図などを用いて説明する。

【0135】まず、リンクコア20の動作について説明する。

【0136】PHYチップを介して他のノードからのパケットを受信すると、パケット診断回路142がそのパケットを診断する。そして、ヘッダ&トレイラー生成回路164がヘッダを生成（整形）する。このヘッダは、バッファ168を介してセクタ170に入力され、パケット診断回路142からの信号SELに基づきセクタ170がこのヘッダを選択する。これにより、図23のA1に示すように、RDとしてヘッダ（H0～H4）がFIFO34に出力されることになる。

【0137】なお、図24（A）に、シリアルバス上で転送される非同期パケットのフォーマット（IEEE1394規格）を示す。一方、図24（B）に、RAM80のヘッダ領域に格納される非同期受信パケットのヘッダ部分のフォーマットを示す（同図において網掛けとなっている部分がトレイラーである）。このように本実施形態では、図24（A）に示すフォーマットのパケットを、ファームウェアなどの上層が使用できるように、図24（B）に示すフォーマットの packets に整形している。

【0138】また本実施形態では、ヘッダの第4クワドレットであるH4（図23のA2）は、図24（B）に示すようにデータ領域からデータを取り出すためのデータポインタになっている。このデータポインタ（H4）は、DMAC44（ポインタ更新回路184）からバッファ168を介してセクタ170に入力され、セクタ170がこれを選択する。このように、パケット整形回路160は、DMAC44からデータポインタを受け取り、そのデータポインタを、RAM80に書き込まれるヘッダに埋め込んでいる。

【0139】次に、PHYチップから、データバスDを介してパケットのデータ部分が送られてくる。直列・並列変換回路132は、このデータ部分を32ビットのデータであるDIに変換し、パケット診断回路142及びバッファ168に出力する。

【0140】なお、DIEは、DIが有効か無効かを示す信号であり、DISは、DIの取り込みタイミングを知らせるためのストロブ信号である。

【0141】直列・並列変換回路132からのDIは、バッファ168を介してセクタ170に入力され、セクタ170がこれを選択する。これにより、A3に示すように、RDとしてデータD0～DnがFIFO34に出力される。

【0142】次に、ヘッダ&トレイラー生成回路164からのトレイラーがバッファ168を介してセクタ170に入力され、セクタ170がこれを選択する。これにより、A4に示すように、RDとしてトレイラー（H5。図24（B）の網掛け部分）がFIFO34に出力される。

【0143】さて、TAG生成回路162は、RDとして出力される情報を区別するためのTAGを生成している。本実施形態では図15に示すようにTAGは2ビットであり、（00）、（01）、（10）、（11）は、各々、ヘッダ、トレイラー、データ、スタート（ヘッダの最初）を表す。従って、例えば図23では、（11）、（00）、…、（10）、…、（01）というようにTAGが変化する。FIFO34には、この2ビットのTAGと32ビットのRDとからなる34ビットのデータが入力されることになる。

【0144】次に、FIFO34の動作について説明する。

【0145】FIFO34は、リンクコア20からのTAG、RDを受け、A5、A6に示すように、DTAG、WDATAとして出力する。

【0146】FIFO34内のFIFO状態判断回路35は、内蔵するカウンタにより、FIFO34のデータ数（FIFOカウント）をカウントする。そして、FIFO34がフル（データ数＝2）になった場合には、図23のA7に示すようにFULLをアクティブ（Hレベル）にする。また、FIFO34が空（データ数＝0）になった場合には、A8に示すようにEMPTYをアクティブにする。FIFO34がフルになったことは、FULL、FFULLにより、DMAC44内のアクセス要求実行回路190やリンクコア20内のシーケンサ167に伝えられる。また、FIFO34が空になったことは、EMPTYにより、DMAC44内のアクセス要求発生回路192に伝えられる。

【0147】次に、DMAC44の動作について説明する。

【0148】アクセス要求発生回路192は、A9に示すようにEMPTYが非アクティブ（Lレベル）になったこと（FIFO34が空でないこと）を条件に、A10に示すようにWREQをアクティブにする。そして、バッファマネージャ70からWACKを受け取ると、WREQを非アクティブにする。

【0149】さて、本実施形態では、受信時のバス調停においては、DMAC44からのアクセス要求の優先順位を最も高くしている。従って、A10、A11に示す

ように、DMAC 44からのWREQと、CPUインターフェース60やポート用のDMAC 54からのOtherWREQとが競合した場合には、WREQの方が優先する。即ち、A12、A13に示すように、WACKの方がOtherWACKよりも先にアクティブになる。このように、WREQとOtherWREQが競合した場合に、WREQの方を優先させるのは、以下の理由による。即ち、IEEE 1394では、SCSIなどとは異なり、他のノードからのパケットが転送クロックに同期して次々に転送されてくる。従って、これらの間断なく転送されてくるパケットを優先してRAM80に次々に格納する必要があるからである。

【0150】また、本実施形態では、CPUインターフェース60やポート用のDMAC 54からのアクセス要求をバッファマネージャ70が受け付けている間は、A14に示すように、DMAC 44のアクセス要求は所与の期間だけ待たされる。従って、リンクコア20からのRDと、バッファマネージャ70へのWDATAとは同期しない。この理由により、本実施形態では、RDとWDATAの位相を調整するためのFIFO 34を設けている。この場合、FIFO 34は、位相調整のために必要な最低限の段数（好ましくは3段以下、更に好ましくは2段以下）を備えていけばよい。

【0151】パケット分離回路180に含まれるTAG判別回路182は、WDATAと共にFIFO 34から出力されるDTAGを判別し、WDATAが、スタート（ヘッダの最初）、ヘッダ、データ、トレイラーのいずれなのかを判定する。そして、ポインタ更新回路184は、この判定結果に基づいて、ヘッダポインタやデータポインタの更新を行う。次に、アドレス発生回路188は、更新されたヘッダポインタやデータポインタに基づいて、WDATAの書き込みアドレスであるWADRを発生する。

【0152】より具体的には、例えば、WDATAがスタート又はヘッダであるとDTAGに基づき判定された場合は、ポインタ更新回路184が、図25（A）に示すように、ヘッダポインタHPのインクリメント（広義には更新）を行う。そして、アドレス発生回路188は、図23のA15に示すように、インクリメントされるヘッダポインタに応じたWADRを発生する。

【0153】次に、WDATAがデータであるとDTAGに基づき判定された場合は、ポインタ更新回路184が、図25（B）に示すように、データポインタDPのインクリメントを行う。このデータポインタDPは、パケット整形回路160によりヘッダの第4クワドレットに埋め込まれたH4に相当する。アドレス発生回路188は、図23のA16に示すように、インクリメントされるデータポインタに応じたWADRを発生する。

【0154】次に、WDATAがトレイラーであるとDTAGに基づき判定された場合は、ポインタ更新回路1

84が、図25（C）に示すように、ヘッダポインタのインクリメントを行う。そして、アドレス発生回路188は、図23のA17に示すように、インクリメントされるヘッダポインタに応じたWADRを発生する。

【0155】なお、最終的には図25（D）に示すように、ヘッダポインタは、処理対象となったパケットのヘッダ部分の下側の境界（次のパケットのヘッダ部分の上側の境界）を指すようになる。また、データポインタは、パケットのデータ部分の下側の境界（次のパケットのデータ部分の上側の境界）を指すようになる。ヘッダポインタやデータポインタの最終位置は、受信に失敗がなかった（RFAILが非アクティブ）ことを条件に、図6のレジスタ46内のヘッダポインタ設定レジスタやデータポインタ設定レジスタにリストアされる。

【0156】以上のようにして、パケットを分離してヘッダ領域とデータ領域に書き込むことが可能になる。

【0157】なお、ヘッダ領域とデータ領域の境界などの、RAM80を分離する領域の境界（図16のP1～P6）の設定は、CPUインターフェース60を介してCPU 66（ファームウェア等）が、図6のレジスタ46に含まれるポインタ設定レジスタに対して、境界のアドレスを指すポインタを設定することで実現される。

【0158】また、データ領域が複数の領域に分離される場合（例えば図16に示すようにアイソクロナス転送用と非同期転送用の領域に分離される場合、或いは第1、第2の非同期転送用の領域に分離される場合等）には、例えばtcodeなどのパケットの制御情報に基づき、分離された複数の領域のいずれかにパケットのデータを書き込むようにすることが望ましい。

【0159】より具体的には、図22に示すように、DMAC 44が、複数のデータポインタ、例えば第1、第2のデータポインタをパケット整形回路160に対して渡すようにする（3個以上のデータポインタを渡してもよい）。そして、パケット整形回路160は、例えば、アイソクロナス転送時（或いは第2の非同期転送時）にはDMAC 44からの第1のデータポインタを選択し、非同転送時（或いは第1の非同期転送時）にはDMAC 44からの第2のデータポインタを選択するようにする。即ち、パケット整形回路160内のパケット診断回路142が、tcodeなどのパケットの制御情報に基づいて、アイソクロナス転送か非同期転送か（或いは第2の非同期転送か第1の非同期転送か）を判別し、この判別結果に基づき信号SELを制御する。そして、バッファ168を介してセクタ170に入力される第1、第2のデータポインタのいずれかを選択するようにする。これにより、アイソクロナス転送（或いは第2の非同期転送）のパケットには第1のデータポインタが埋め込まれ、非同期転送（或いは第1の非同期転送）のパケットには第2のデータポインタが埋め込まれるようになる。この結果、データ領域を分離する特定の領域に、連

続的にデータを格納することが可能になる。即ち、デジタルカメラにおける動画データを送信するデータ領域に連続的に格納したり、プリンタにおける印字データを第2の非同期転送用データ領域に連続的に格納したりすること（第1の非同期転送用データ領域にはコマンドデータ、ステータスデータなどの制御用データを格納する）が可能になる。

【0160】また、本実施形態では図24(B)に示すように、ヘッダの後ろ側に、spd（データ転送の速度を特定するためのスピードコード情報）、BC（バスに接続される全てのノードに対して送信されたパケット可否を示すブロードキャスト情報）、HCE（パケットにエラーがあったか否かを示すエラーステータス情報）、ACK（ソースノードに返したアックノリジメント情報）が付加される。これらのspd、BC、HCE、ACKは、図22のリンクコア20内のヘッダ&トレーラー生成回路164により生成され、バッファ168、セクタ170の機能により付加される。

【0161】また、図22のエラーチェック回路166は、受信パケットのヘッダCRCをチェックし、受信パケットのヘッダにエラーがあった場合には、HCEを1に設定する。そして、HCEが1になると、RFAILがアクティブになる。そして、RFAILがアクティブになると、データポインタの最終位置（図25(D)のH4+(n+1)）は、図6のレジスタ46内のデータポインタ設定レジスタにリストアされない。これにより、図18(B)で説明した、データポインタDPを元に戻しデータ3を無効にする処理が実現される。

【0162】また、図26(A)に、パケットがセルフIDパケットである場合に、RAM80のヘッダ領域に格納されるヘッダ部分のフォーマットを示し、図26(B)に、パケットがセルフIDパケット以外のPHYパケットである場合に、ヘッダ領域に格納されるヘッダ部分のフォーマットを示す。

【0163】パケットがセルフID期間中のセルフIDパケットである場合には、図26(A)に示すように、セルフID期間に受信したパケットか否かを示す情報であるBRが1となる。一方、セルフID期間外のリンクオンパケットやPHY構成パケットである場合には、図26(B)に示すようにBRが0になる。このようにすることで、tcodeが同じ0xEであっても、ファームウェアはこれらのパケットを区別できるようになる。

【0164】なお、図24(B)、図26(A)、

(B)では、spd、BC、HCE、ACK、BRを、ヘッダの後ろ側に付加しているが、図21(B)で説明したように、これらのspd、BC、HCE、ACK、BRを、ヘッダの先頭側に付加するようにしてもよい。

【0165】4. 電子機器

次に、本実施形態のデータ転送制御装置を含む電子機器の例について説明する。

【0166】例えば図27(A)に電子機器の1つであるプリンタの内部ブロック図を示し、図28(A)にその外観図を示す。CPU（マイクロコンピュータ）510はシステム全体の制御などを行う。操作部511はプリンタをユーザが操作するためのものである。ROM516には、制御プログラム、フォントなどが格納され、RAM518はCPU510のワーク領域として機能する。表示パネル519はプリンタの動作状態をユーザに知らせるためのものである。

【0167】PHYチップ502、データ転送制御装置500を介して、パーソナルコンピュータなどの他のノードから送られてきた印字データは、バス504を介して印字処理部512に直接送られる。そして、印字データは、印字処理部512にて所与の処理が施され、プリントヘッドなどからなる印字部（データを出力するための装置）514により紙に印字されて出力される。

【0168】図27(B)に電子機器の1つであるスキャナの内部ブロック図を示し、図28(B)にその外観図を示す。CPU520はシステム全体の制御などを行う。操作部521はスキャナをユーザが操作するためのものである。ROM526には制御プログラムなどが格納され、RAM528はCPU520のワーク領域として機能する。

【0169】光源、光電変換器などからなる画像読み取り部（データを取り込むための装置）522により原稿の画像が読み取られ、読み取られた画像のデータは画像処理部524により処理される。そして、処理後の画像データがバス505を介してデータ転送制御装置500に直接送られる。データ転送制御装置500は、この画像データにヘッダなどを付加することでパケットを生成し、PHYチップ502を介してパーソナルコンピュータなどの他のノードに送信する。

【0170】図27(C)に電子機器の1つであるCD-Rドライブの内部ブロック図を示し、図28(C)にその外観図を示す。CPU530はシステム全体の制御などを行う。操作部531はCD-Rをユーザが操作するためのものである。ROM536には制御プログラムなどが格納され、RAM538はCPU530のワーク領域として機能する。

【0171】レーザ、モータ、光学系などからなる読み取り&書き込み部（データを取り込むための装置又はデータを記憶するための装置）533によりCD-R532から読み取られたデータは、信号処理部534に入力され、エラー訂正処理などの所与の信号処理が施される。そして、信号処理が施されたデータが、バス506を介してデータ転送制御装置500に直接送られる。データ転送制御装置500は、このデータにヘッダなどを付加することでパケットを生成し、PHYチップ502を介してパーソナルコンピュータなどの他のノードに送信する。

【0172】一方、PHYチップ502、データ転送制御装置500を介して、他のノードから送られてきたデータは、バス506を介して信号処理部534に直接送られる。そして、信号処理部534によりこのデータに所与の信号処理が施され、読み取り&書き込み部533によりCD-R532に記憶される。

【0173】なお、図27(A)、(B)、(C)において、CPU510、520、530の他に、データ転送制御装置500でのデータ転送制御のためのCPUを別に設けるようにしてもよい。

【0174】本実施形態のデータ転送制御装置を電子機器に用いることで、高速なデータ転送が可能になる。従って、ユーザがパーソナルコンピュータなどによりプリントアウトの指示を行った場合に、少ないタイムラグで印字が完了するようになる。また、スキャナへの画像取り込みの指示の後に、少ないタイムラグで読み取り画像をユーザは見るができるようになる。また、CD-Rからのデータの読み取りや、CD-Rへのデータの書き込みを高速に行うことができるようになる。更に、例えば1つのホストシステムに複数の電子機器を接続して利用したり、複数のホストシステムに複数の電子機器を接続して利用したりすることも容易になる。

【0175】また本実施形態のデータ転送制御装置を電子機器に用いることで、CPU上で動作するファームウェアの処理負荷が軽減され、安価なCPUや低速のバスを用いることが可能になる。更に、データ転送制御装置の低コスト化、小規模化を図れるため、電子機器の低コスト化、小規模化も図れるようになる。

【0176】なお本実施形態のデータ転送制御装置を適用できる電子機器としては、上記以外にも例えば、種々の光ディスクドライブ(CDROM、DVD)、光磁気ディスクドライブ(MO)、ハードディスクドライブ、TV、VTR、ビデオカメラ、オーディオ機器、電話機、プロジェクタ、パーソナルコンピュータ、電子手帳、ワードプロセッサなど種々のものを考えることができる。

【0177】なお、本発明は本実施形態に限定されず、本発明の要旨の範囲内で種々の変形実施が可能である。

【0178】例えば、本発明のデータ転送制御装置の構成は、図6に示す構成が特に望ましいが、これに限定されるものではない。特に、ブロードキャスト情報やエラーステータス情報や自己識別期間に受信したパケットが否かを示す情報を、パケットの制御情報に付加する発明や、時系列で後ろ側にある所与の情報をパケットの制御情報の先頭側に書き込む発明では、図6のみならず、図8などの種々の構成を採用することができる。

【0179】また、本発明は、IEEE1394規格でのデータ転送に適用されることが特に望ましいが、これに限定されるものではない。例えばIEEE1394と同様の思想に基づく規格やIEEE1394を発展させ

た規格におけるデータ転送にも本発明は適用できる。

【0180】

【図面の簡単な説明】

【図1】図1(A)、(B)、(C)は、非同期転送とアイソクロナス転送について説明するための図である。

【図2】図2(A)、(B)は、ツリー識別について説明するための図である。

【図3】自己識別について説明するための図である。

【図4】図4(A)、(B)、(C)、(D)は、セルフIDパケットなどの物理層のパケットのフォーマットを示す図である。

【図5】IEEE1394のプロトコル構成について示す図である。

【図6】本実施形態のデータ転送制御装置の構成例を示す図である。

【図7】ヘッダ(制御情報)領域とデータ領域の分離について説明するための図である。

【図8】本実施形態の比較例の構成例について示す図である。

【図9】図8の構成によるデータ転送の手法について説明するための図である。

【図10】データ転送の手法の他の例について説明するための図である。

【図11】本実施形態のデータ転送の手法について説明するための図である。

【図12】受信パケットをヘッダとデータに分離してRAMのヘッダ領域とデータ領域に格納する手法について説明するための図である。

【図13】ヘッダ領域に格納するヘッダにデータポイントを含ませる手法について説明するための図である。

【図14】パケット分離回路が渡したデータポイントを、パケット整形回路がパケットのヘッダに付加する手法について説明するための図である。

【図15】TAGについて説明するための図である。

【図16】RAMの各領域のサイズを可変に制御する手法について説明するための図である。

【図17】図17(A)、(B)は、ブロードキャスト情報をパケットのトレーラーに付加する手法について説明するための図である。

【図18】図18(A)、(B)は、エラーステータス情報をパケットのトレーラーに付加すると共に、パケットにエラーがあった場合にRAMのデータを無効にする手法について説明するための図である。

【図19】セルフID期間について説明するための図である。

【図20】図20(A)、(B)は、セルフID期間に受信したパケットが否かを示す情報を、パケットのトレーラーに付加する手法について説明するための図である。

【図21】図21(A)、(B)は、パケット整形にお

いて時系列でパケットの後ろ側に付加された情報を、RAMにおいてパケットのヘッダの先頭側に書き込む手法について説明するための図である。

【図22】受信側の構成の一例を示す図である。

【図23】受信側の動作について説明するためのタイミング波形図である。

【図24】図24(A)は、IEEE1394規格の非同期パケットのフォーマットであり、図24(B)は、RAMのヘッダ領域に格納される非同期受信パケットのヘッダ部分のフォーマットである。

【図25】図25(A)、(B)、(C)、(D)は、ヘッダポインタとデータポインタの更新について説明するための図である。

【図26】図26(A)は、パケットがセルフIDパケットである場合のヘッダ部分のフォーマットであり、図26(B)は、パケットがセルフIDパケット以外のPHYパケットである場合のヘッダ部分のフォーマットである。

【図27】図27(A)、(B)、(C)は、種々の電子機器の内部ブロック図の例である。

【図28】図28(A)、(B)、(C)は、種々の電子機器の外観図の例である。

【符号の説明】

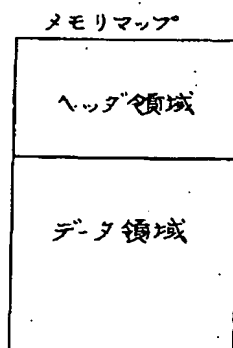
- 10 PHYインターフェース
- 20 リンクコア
- 22 レジスタ
- 30 FIFO(ATF)
- 32 FIFO(ITF)
- 34 FIFO(RF)
- 40 DMAC(ATF用)
- 42 DMAC(ITF用)
- 44 DMAC(RF用)
- 50 ポートインターフェース
- 52 FIFO(PF)
- 54 DMAC(PF用)
- 56 レジスタ

*

- *60 CPUインターフェース
- 62 アドレスデコーダ
- 63 データ同期化回路
- 64 割り込みコントローラ
- 66 CPU
- 68 クロック制御回路
- 70 バッファマネージャ
- 72 レジスタ
- 74 調停回路
- 10 76 シーケンサ
- 80 RAM(パケット記憶手段)
- 90、92、94 バス(第1のバス)
- 96、98 バス(第2のバス)
- 100、102、104、105、
- 106、107、108、109 バス(第3のバス)
- 110 バス(第4のバス)
- 120 データ転送制御装置
- 122 PHYチップ
- 124 アプリケーション層のデバイス
- 20 130 バス監視回路
- 132 直列・並列変換回路
- 142 パケット診断回路
- 160 パケット整形回路
- 162 TAG生成回路
- 164 ヘッダ&トレイラー生成回路
- 166 エラーチェック回路
- 167 シーケンサ
- 168 バッファ
- 170 セレクタ
- 30 180 パケット分離回路
- 182 TAG判別回路
- 184 ポインタ更新回路
- 188 アドレス発生回路
- 190 アクセス要求実行回路
- 192 アドレス要求発生回路

*

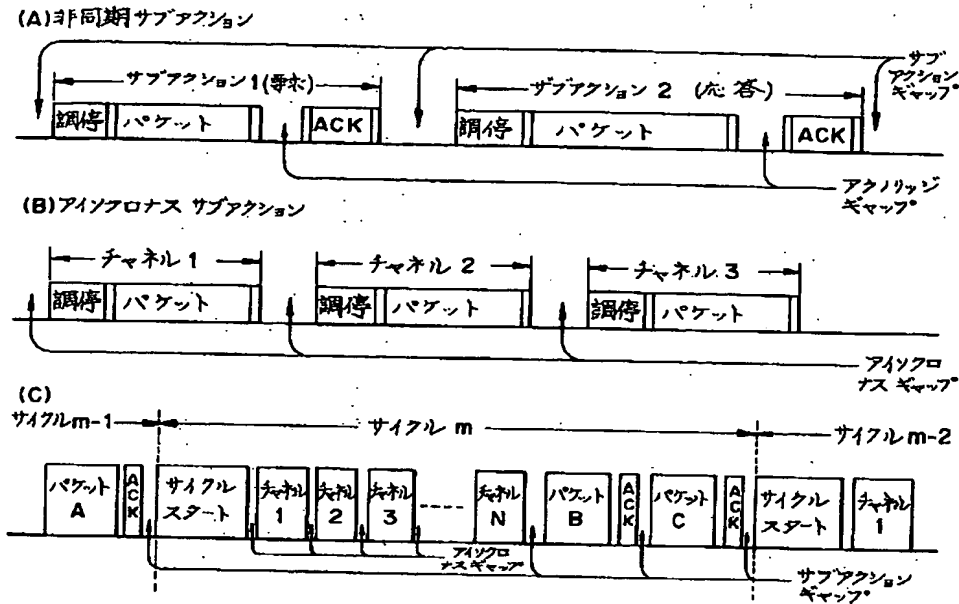
【図7】



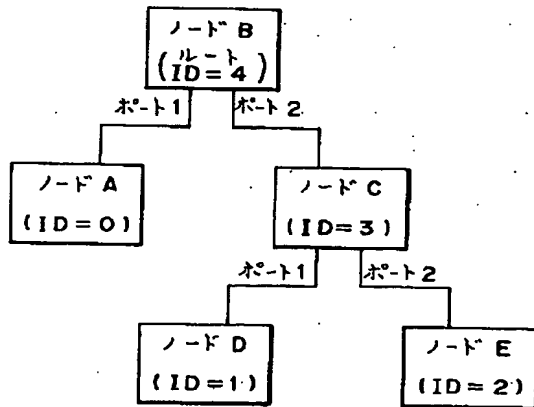
【図15】

TAG(DTAG)	意味
00	ヘッダ
01	トレイラー
10	データ
11	スタート

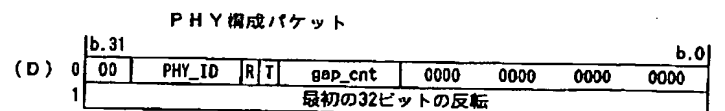
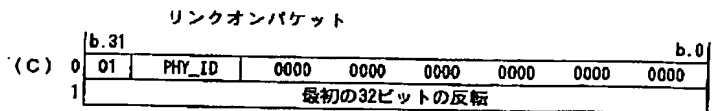
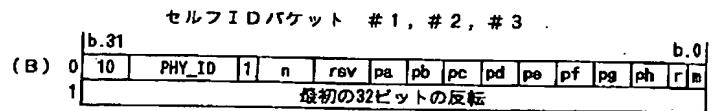
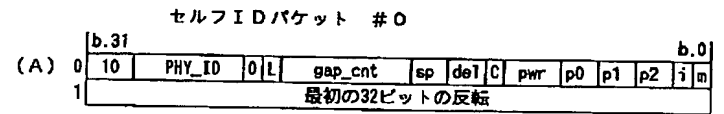
【図1】



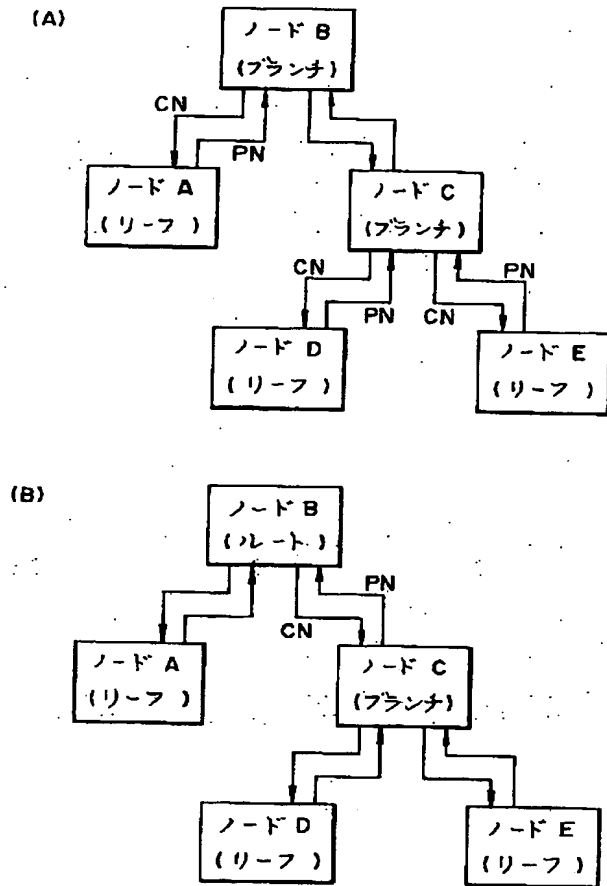
【図3】



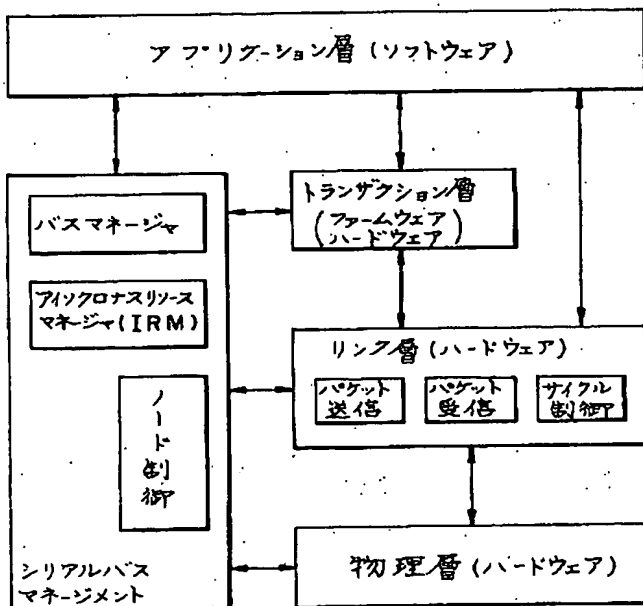
【図4】



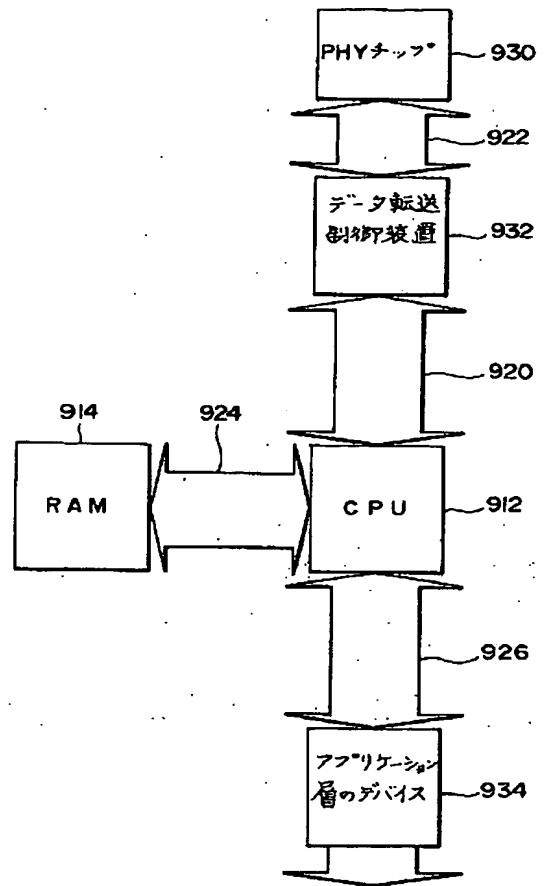
【図2】



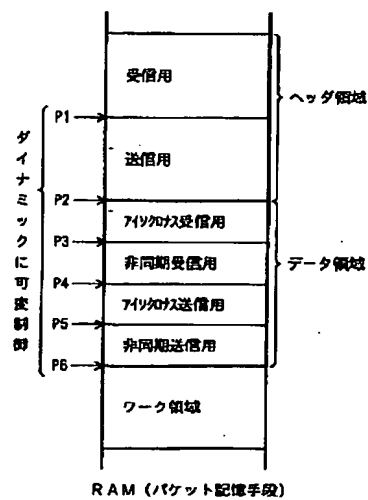
【図5】



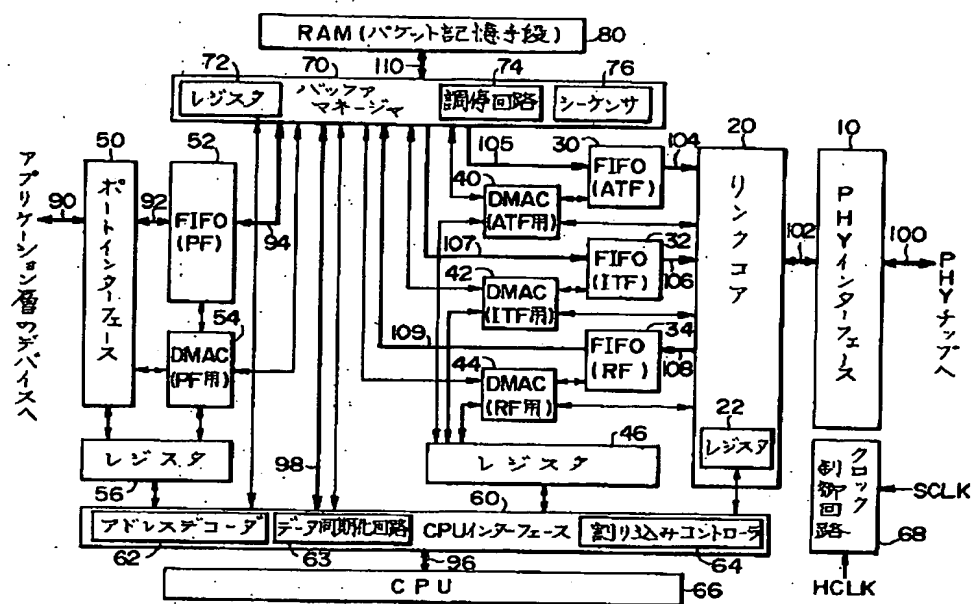
【図9】



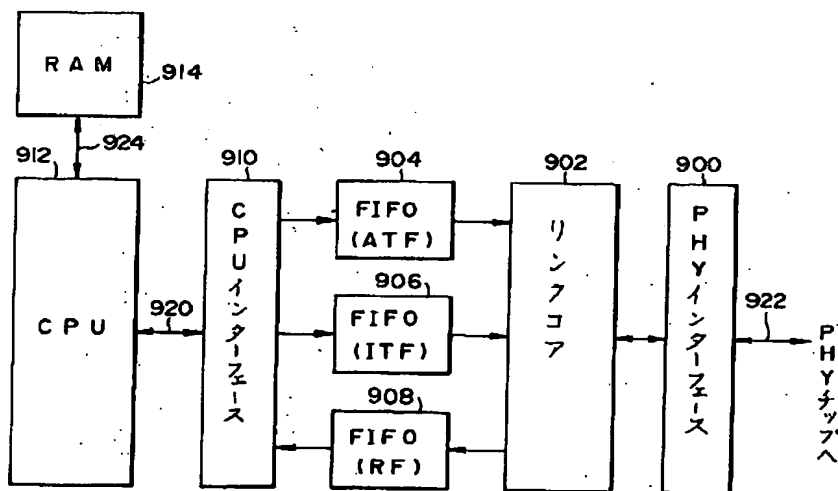
【図16】



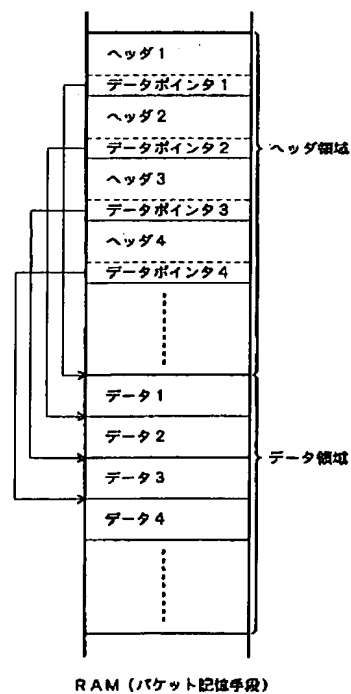
【図6】



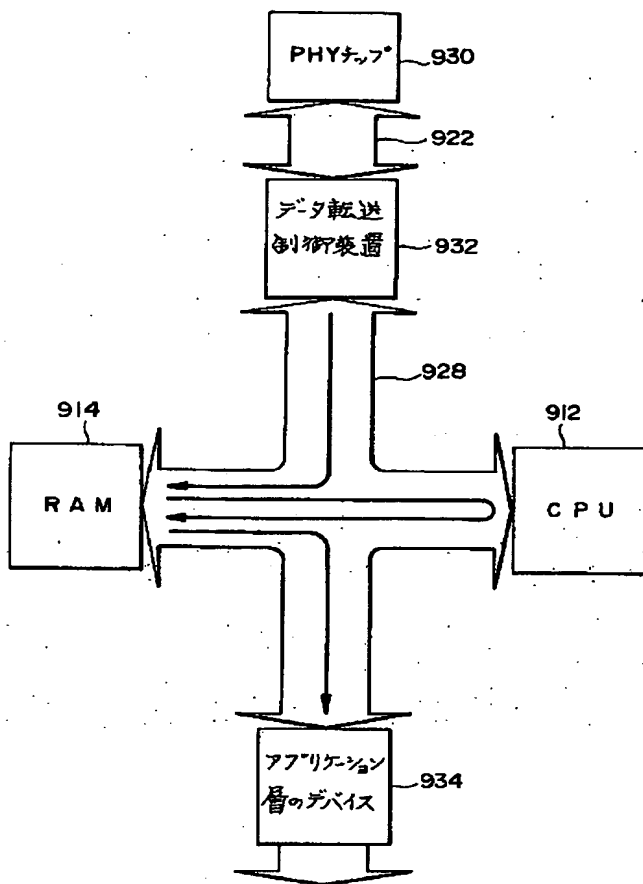
【図8】



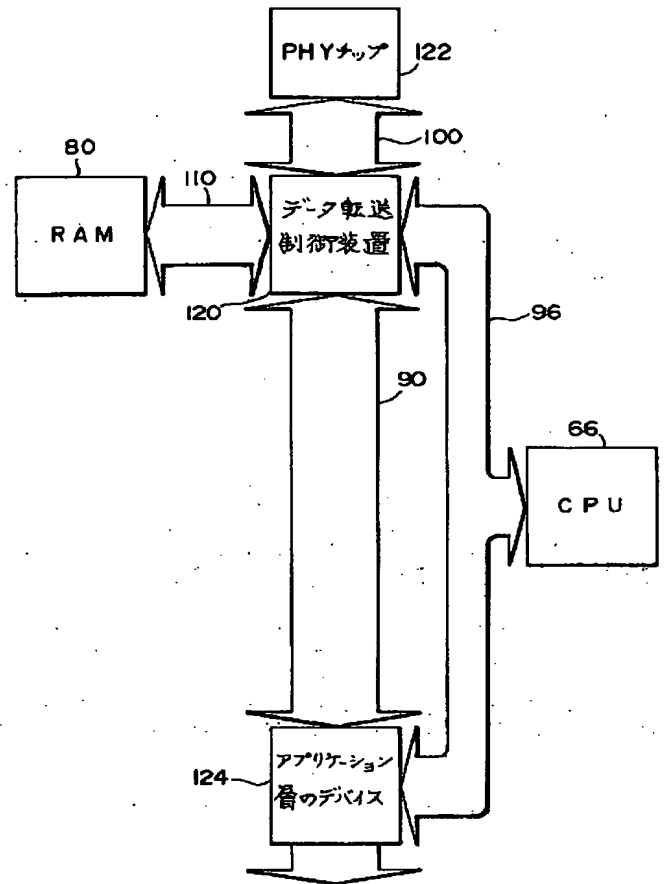
【図13】



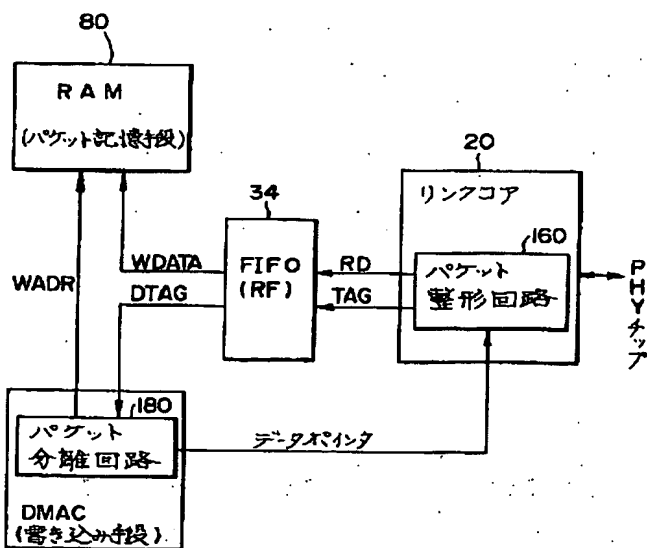
【図10】



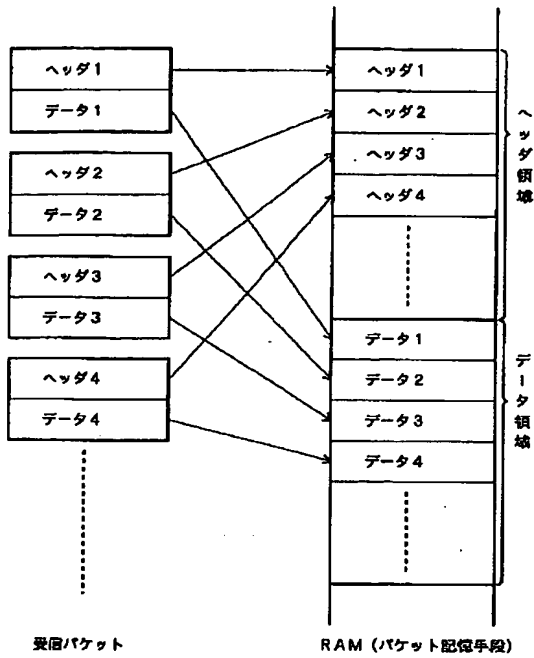
【図11】



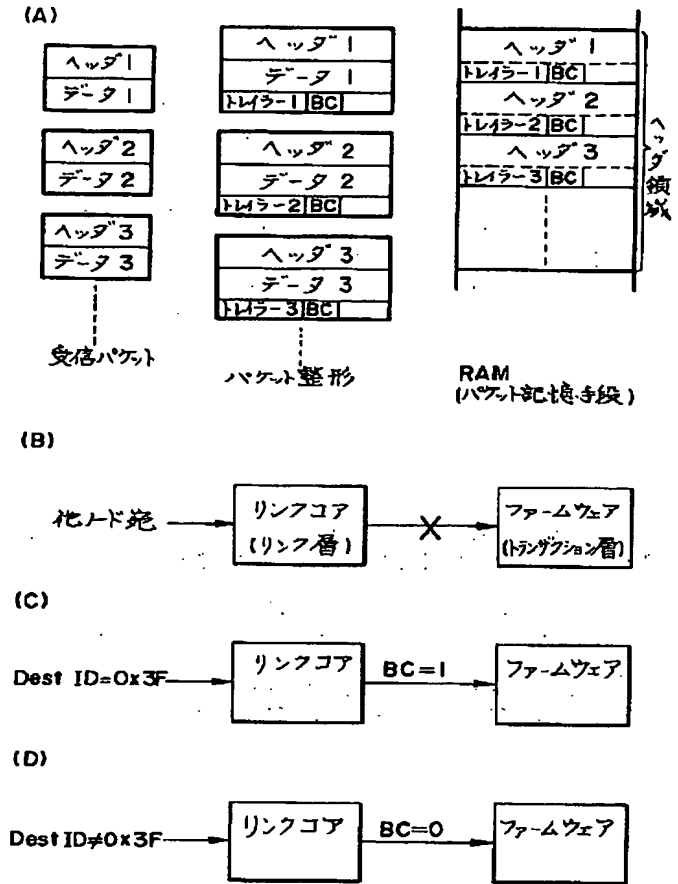
【図14】



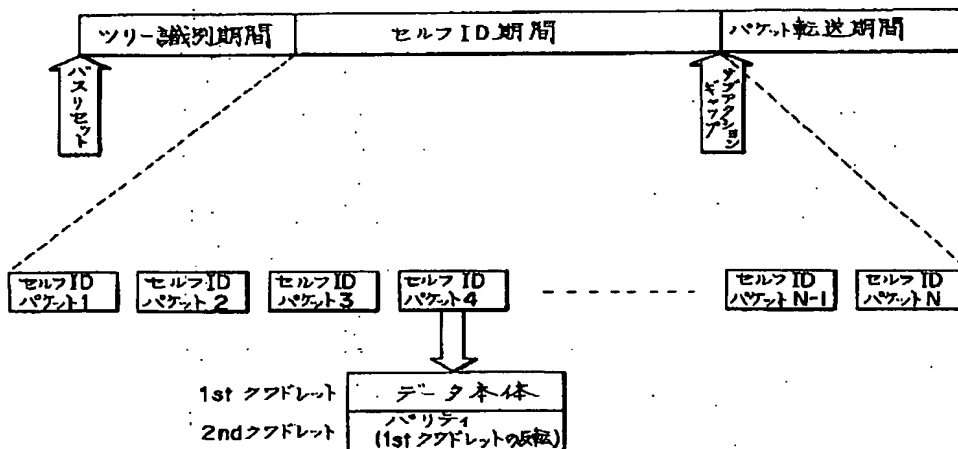
【図12】



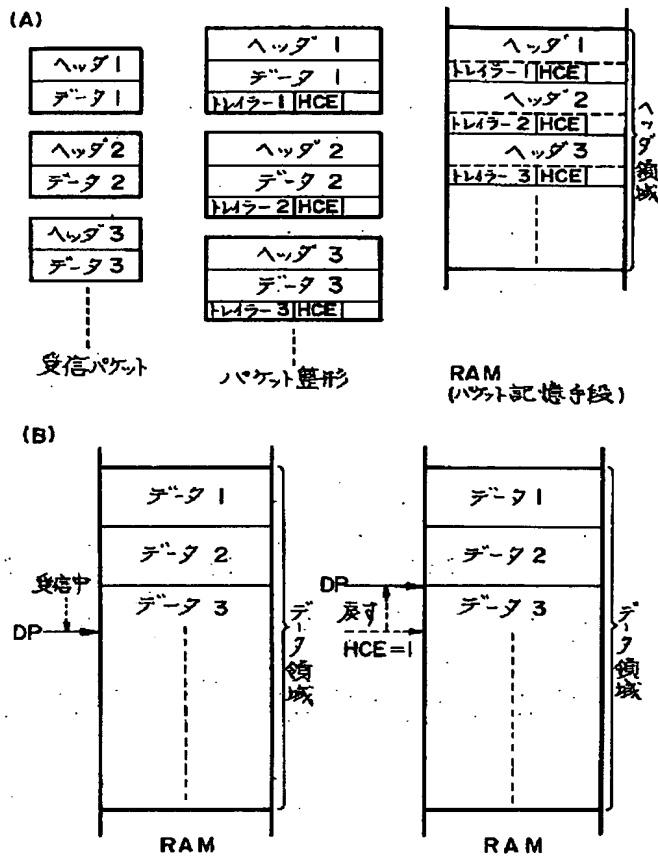
【図17】



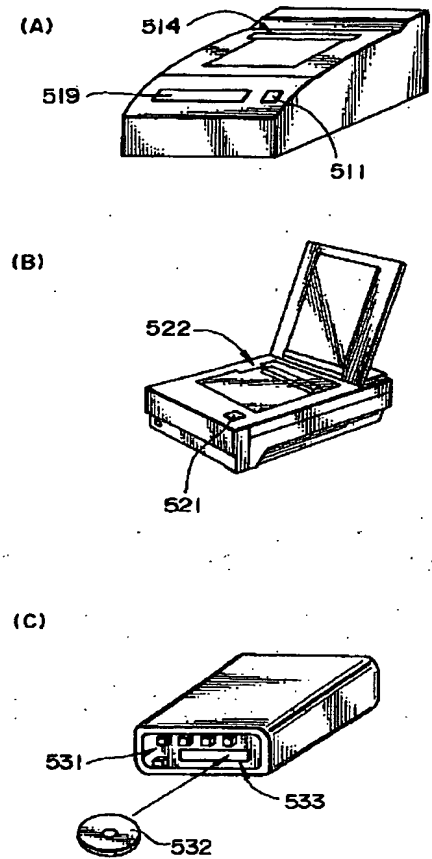
【図19】



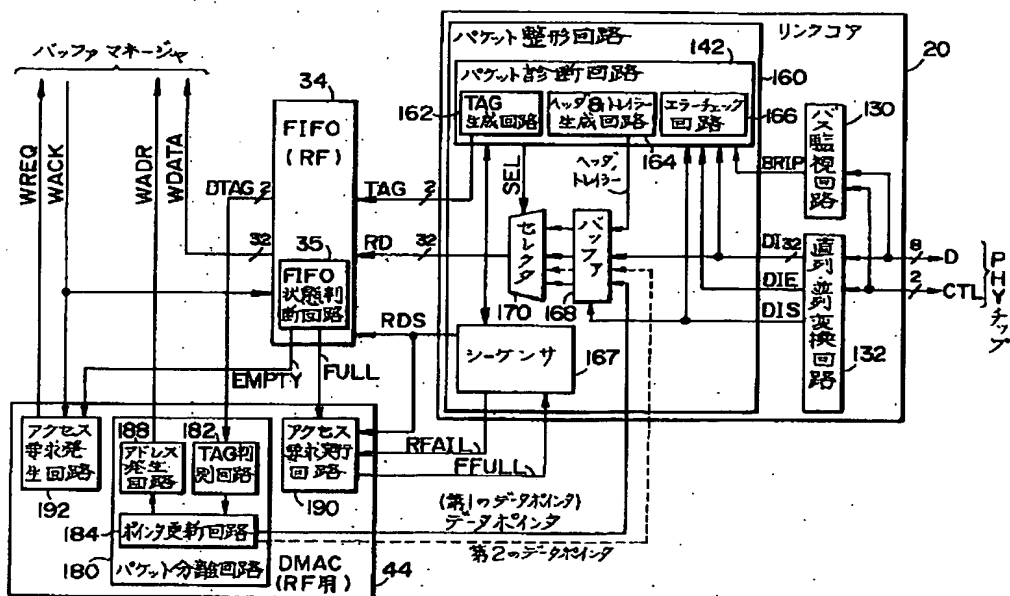
【図18】



【図28】

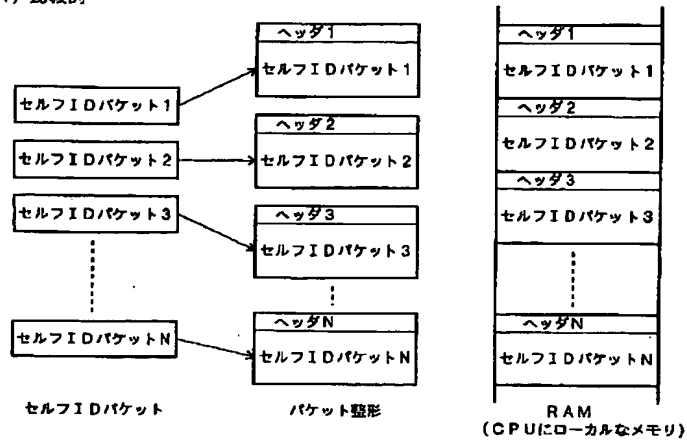


【図22】

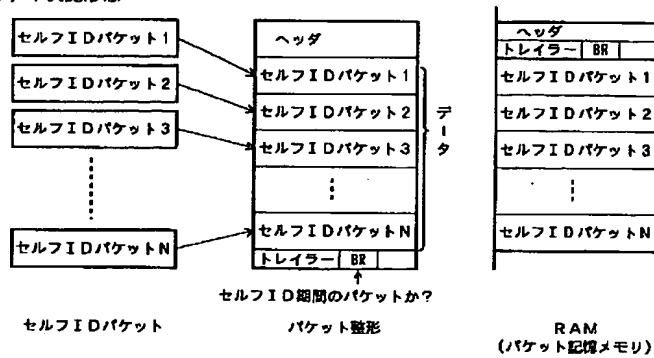


【図20】

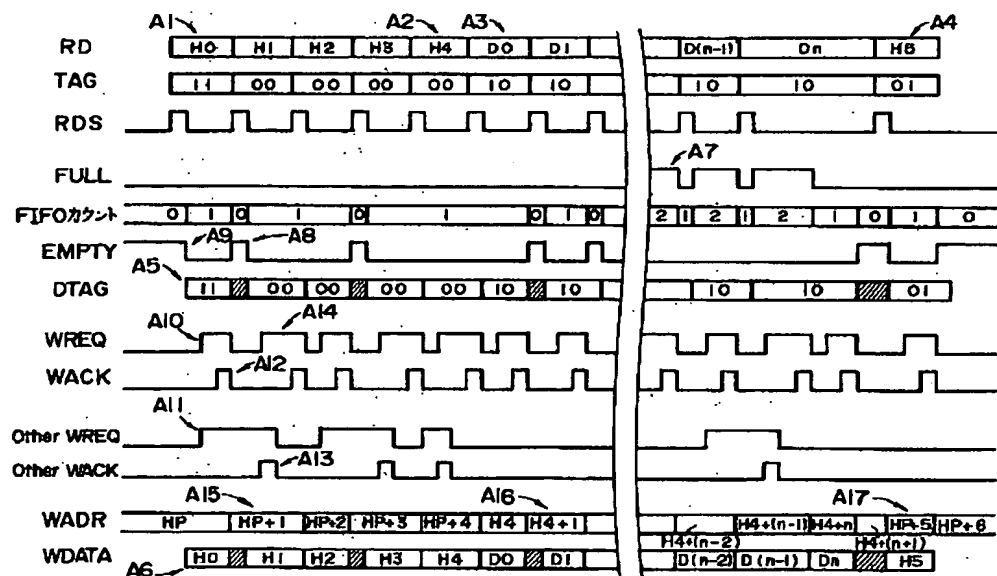
(A) 比較例



(B) 本実施形態

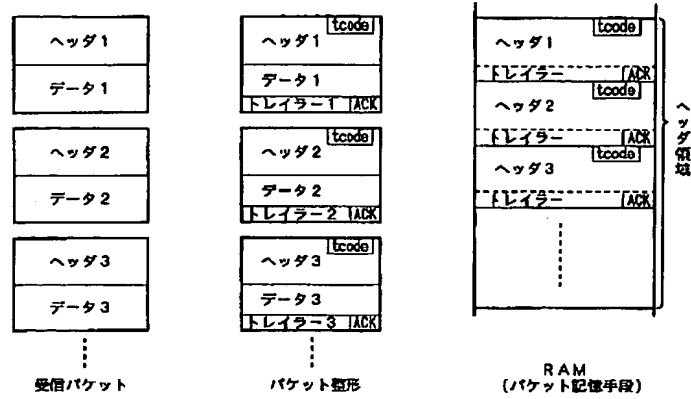


【図23】

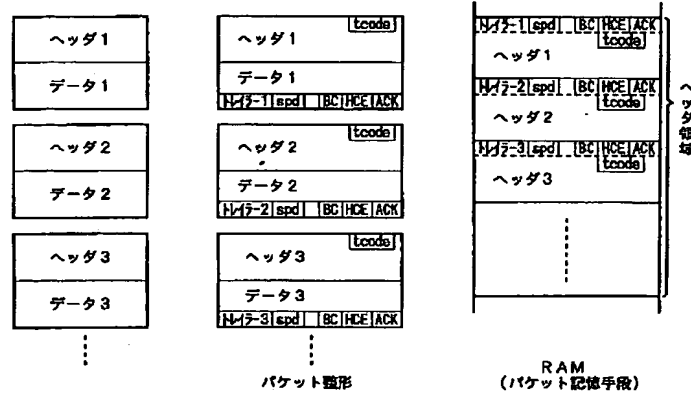


【図 2 1】

(A)

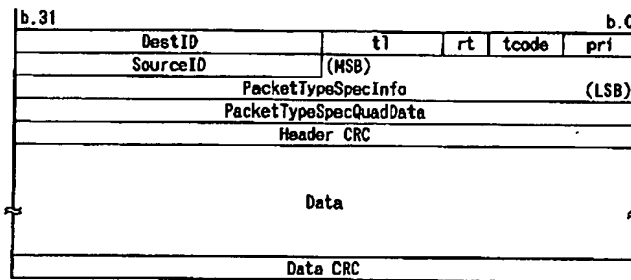


(B)

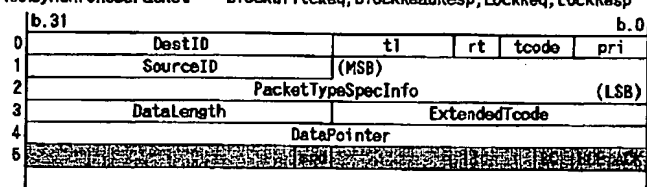


【図 2 4】

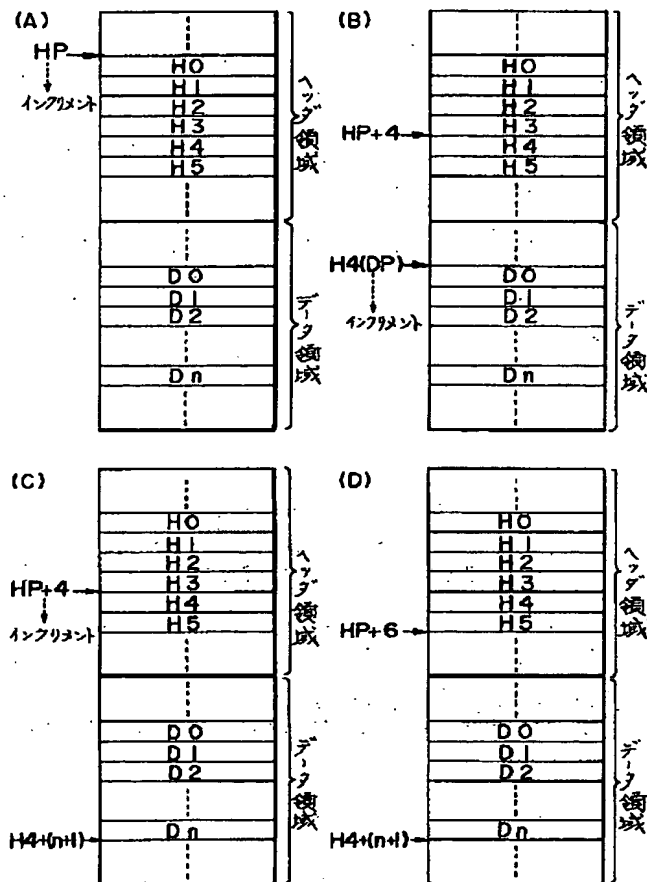
(A)



(B) RxAsynchronousPacket BlockWriteReq, BlockReadResp, LockReq, LockResp

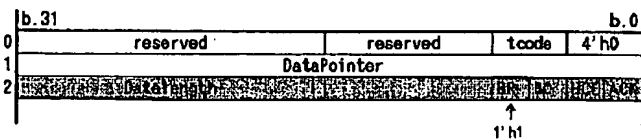


【図25】

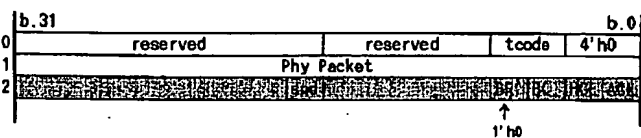


【図26】

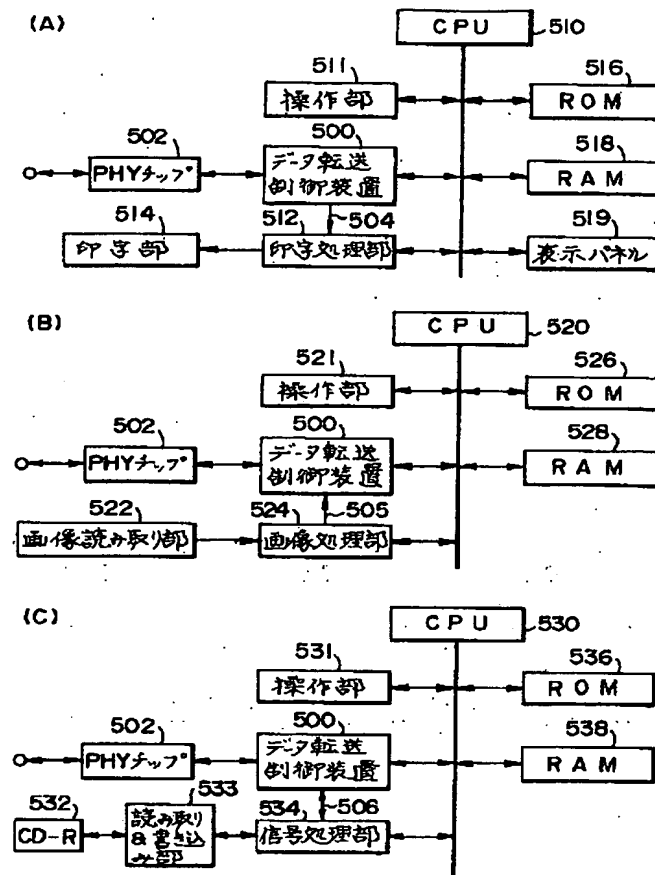
(A) RxAsynchronousBusResetPacket (tcode:0xE)



(B) RxAsynchronousPhyPacket Normal (tcode:0xE)



【図27】



フロントページの続き

(72)発明者 和田 文利
 長野県諏訪市大和3丁目3番5号 セイコ
 ーエプソン株式会社内

Fターム(参考) 5K032 AA02 AA03 CC03 CC10 CD01
 DB19 DB24